fare elettronica

CULTURA ELETTRONICA APPLICATA

130 pagine!

NOVEMBRE
n° 245 • Anno 21

Interfaccia IR per PC

Controllare il PC con un normale telecomando

Introduzione alle memorie Flash

Mikrobasic
Pilotiamo i motori passo-passo

La tecnica DDS
Riproduttore di suonerie per cellulari

Usare Proteus
ISIS: creazione dei componenti ed editing avanzato

Alimentatori Switching
Switching flyback multi-uscita

Vitamina C
Gestione delle interruzioni

ELETTRONICANDO
Il timer 555

INWARE EDIZIONI
ISSN 1591-2272
9 771591 227008 50245

€ 5,50

Poste Italiane Spa - Spedizione in abbonamento Postale - D.L. 353/2003 (conv. In L. 27/02/2004 n. 46) Art. 1 comma 1 - CNS Milano
Caso di mancato recapito, restituire all'editore che si impegna a pagare la relativa tassa presso il CPM di Roserio - Milano

Un PC amichevole

Qualche tempo fa ho comprato, per mia figlia di quattro anni, una bellissima raccolta di favole su CD, lei ne è entusiasta e non perde occasione per ascoltarle utilizzando il suo piccolo lettore CD. Purtroppo ha fatto i conti senza l'oste, una peste di quasi due anni che ha come unico scopo nella vita quello di giocare con i suddetti CD che, al contrario di quanto si possa pensare, sono sensibili ai graffi e all'impeto artistico del piccolo pittore, soprattutto se utilizza pennarelli indelebili. Ho dovuto quindi affannarmi a cercare una soluzione. La prima cosa a cui ho pensato è un lettore MP3, ma questi dispositivi presentano due problemi che ne limitano l'utilizzo da parte dei bambini: è possibile ascoltare la musica solo attraverso la cuffia e dispongono di tasti troppo piccoli da manovrare agevolmente. Ho deciso quindi di utilizzare un vecchio PC che avevo in soffitta, ho installato una distribuzione Linux mirata a creare un PC multimediale, ho convertito le favole in formato MP3 ed i cartoni animati (su DVD) in formato DivX, risolvendo così il problema dei graffi su questi costosi supporti. Ho nascosto il PC, lasciando a portata di mano solo la tastiera, e l'ho collegato al TV. Ma anche io non ho fatto i conti con il piccolo vandalo, il quale, defraudato della possibilità di distruggere i cd, ha concentrato tutte le sue attenzioni sulla tastiera...
La soluzione definitiva è arrivata quando *Enrico Raffone*, nostro valido autore, mi ha proposto un articolo (presentato in questo numero) sul progetto *LIRC*, volto appunto ad interfacciare un qualsiasi telecomando al PC, sia che utilizzi Linux sia Windows. Grazie a questo articolo ho realizzato l'interfaccia IR ed utilizzato il telecomando di un vecchio televisore ormai dismesso. Questa soluzione mi ha consentito di nascondere la tastiera, lasciando il piccolo vandalo a bocca asciutta. Lui però non si è perso d'animo, rivolgendo le sue attenzioni distruttive sulla mia *preziosissima* raccolta di CD audio, ma questa è un'altra storia...
Con una spesa bassissima ed un po' di pazienza ho reso un PC *amichevole*, vi consiglio quindi la realizzazione di questo progetto, è davvero semplice, utile e divertente.

Come potete notare, anche in questo numero abbiamo aggiunto ben 16 pagine di contenuti in più, tutti di rilievo. Si va dalle nuove puntate di *Elettronicando*, che questo mese è dedicata ad un componente storico - il timer 555, ad *Alimentatori Switching*, *Vitamina C*, *Amplificatori operazionali dalla A alla Z*, *CPLD By Example*, *Usare PROTEUS*, fino alla nuova puntata del corso sul *Mikrobasic*, dedicata alla gestione dei motori passo-passo, cosa che farà sicuramente piacere agli appassionati di robotica. Come dicevo l'articolo centrale in questo numero è *l'interfaccia IR per PC*, ma particolare attenzione merita anche un curioso progetto della serie dedicata alla *tecnica DDS*: un *riproduttore di suonerie per cellulari*. Chiude il numero uno speciale che vi consiglio vivamente di leggere, in quanto prima o poi avrete a che fare con l'argomento trattato: le *memorie Flash*.

Prima di concludere, vi ricordo la presenza in edicola, a partire dal 15 Novembre, del numero speciale: **100 idee di progetto per 1000 applicazioni**. Questo speciale, in omaggio agli abbonati, è una raccolta di soluzioni circuitali a tanti problemi elettronici che sicuramente vi tornerà utile. Prenotate la vostra copia per non rischiare di perderla!

Vi auguro quindi una piacevole lettura e vi rinnovo l'appuntamento in edicola a Dicembre.

Tiziano Galizia
t.galizia@farelettronica.com

Fare elettronica
CULTURA ELETTRONICA APPLICATA
www.farelettronica.com

DIRETTORE RESPONSABILE
Antonio Cirella

DIRETTORE DI REDAZIONE
Tiziano Galizia

DIRETTORE TECNICO
Maurizio Del Corso

HANNO COLLABORATO IN QUESTO NUMERO
Giovanni Di Maria, Riccardo Nicoletti, Teodoro Fristachi, Salvatore Torrisi, Enrico Raffone, Antonio Di Stefano, Romano Bernarducci, Nico Grilloni, Agostino Rolando.

DIREZIONE • REDAZIONE • PUBBLICITÁ
INWARE srl - Via Cadorna, 27/31 - 20032 Cormano (MI)
Tel. 02.66504794 - 02.66504755 - Fax 02.66508225
info@inware.it - www.inwaredizioni.it
Redazione: redazione@farelettronica.com

PROGETTO GRAFICO E IMPAGINAZIONE
Graficonsult - Milano

STAMPA
ROTO 2000 - Via L. da Vinci, 18/20 - 20080, Casarile (MI)

DISTRIBUZIONE
Parrini & C. S.p.a. - Viale Forlanini, 23 - 20134, Milano

UFFICIO ABBONAMENTI
INWARE srl - Via Cadorna, 27/31 - 20032 Cormano (MI)
Per informazioni, sottoscrizione o rinnovo dell'abbonamento:
abbonamenti@farelettronica.com
Tel. 02.66504794 - Fax. 02.66508225
L'ufficio abbonamenti è disponibile telefonicamente dal lunedì al venerdì dalle 14,30 alle 17,30

Poste Italiane S.p.a. - Spedizione in abbonamento Postale
D.L. 353/2003 (conv. In L. 27/02/2004 n. 46) art. 1, comma1, DCB Milano.
Abbonamento per l'Italia: € 45,00
Abbonamento per l'estero: € 115,00

Gli arretrati potranno essere richiesti, per iscritto, al seguente costo:
Numero singolo: € 7,50
Numero doppio: € 9,00
Numero con allegato: € 8,50

Autorizzazione alla pubblicazione del Tribunale di Milano n. 647 del 17/11/2003.
Iscrizione al R.O.C. n. 11035 19/11/2003

© Copyright - Tutti i diritti di riproduzione o di traduzione degli articoli pubblicati sono riservati. Manoscritti, disegni e fotografie sono di proprietà di INWARE srl. È vietata la riproduzione anche parziale degli articoli salvo espressa autorizzazione scritta dell'editore. I contenuti pubblicitari sono riportati senza responsabilità, a puro titolo informativo.

Privacy - Nel caso la rivista sia pervenuta in abbonamento o in omaggio, si rende noto che i dati in nostro possesso sono impiegati nel pieno rispetto del D.Lgs. 196/2003. I dati trasmessi a mezzo cartoline o questionari presenti nella rivista, potranno venire utilizzati per indagini di mercato, proposte commerciali, o l'inoltro di altri prodotti editoriali a scopo di saggio. L'interessato potrà avvalersi dei diritti previsti dalla succitata legge. In conformità a quanto disposto dal Codice di deontologia relativo al Trattamento di dati personali art. 2, comma 2, si comunica che presso la nostra sede di Cormano Via Cadorna 27, esiste una banca dati di uso redazionale. Gli interessati potranno esercitare i diritti previsti dal D.Lgs. 196/2003 contattando il Responsabile del Trattamento Inware Srl (info@inwaredizioni.it).

RICHIESTE DI ASSISTENZA
Per richiedere assistenza o chiarimenti sugli articoli pubblicati, vi preghiamo di contattare direttamente l'autore. Se questo non fosse possibile, utilizzate il modulo di contatto che trovate sul nostro sito web www.farelettronica.com.

COLLABORARE CON FARE ELETTRONICA
Le richieste di collaborazione vanno indirizzate all'attenzione di Tiziano Galizia *(t.galizia@farelettronica.com)* e accompagnate, se possibile, da una breve descrizione delle vostre competenze tecniche e/o editoriali, oltre che da un elenco degli argomenti e/o progetti che desiderate proporre.

ELENCO INSERZIONISTI

Alterlogix ..... **pag. 23**
Via Giotto, 19 - 64026 Roseto degli Abruzzi (TE)
Tel 0858933615 - www.alterlogix.com

Artek Electronics Solution ..... **pag. 7**
Piazza Pirazzoli, 2 - 40020 Sasso Morelli (BO)
Tel 0542.643192 - www.artek.it

Blu Nautilus ..... **pag. 21**
Piazza Tre Martiri, 24 - 47900 Rimini
Tel 0541.53294 - www.blunautilus.it

Compendio Fiere ..... **pag. 17-45**
Via Mammianese, 53 - 51017 Pescia (PT)
Tel 0572.490868 - www.compendiofiere.it

Comprel ..... **pag. 89**
Via Saragat, 4 - 20054 Nova Milanese (MI)
Tel 0572521000 - www.comprel.it

Consorzio Elettrimpex. ..... **pag. 93**
Via Console Flaminio,19 - 20134 Milano
Tel 02.210111230 - www.elettrimpex.it

E.R.F ..... **pag. 71**
L.go Fiera della Pesca, 11 - 60125 Ancona
Tel 071.58971 - www.erf.it

E V R Electronics ..... **pag. 13**
Viale Kennedy, 96 - 20027 Rescaldina (MI)
Tel 0331.464893 - www.evr-electronics.com

Futura Elettronica ..... **pagg. 11-41-75-101**
Via Adige, 11 - 21013 Gallarate (VA)
Tel 0331.792287 - www.futuranet.it

Grifo ..... **II cop.**
Via dell'Artigiano, 8/6 - 40016 San Giorgio Di Piano (BO)
Tel 051.892052 - www.grifo.it

Idea Elettronica **37**
Via S. Vittore 24/A - 21040 Oggiona con S.Stefano (VA)
Tel 0331.502868 - www.ideaelettronica.it

Millenium Dataware **67**
Corso Repubblica, 48 - 15057 Tortona (AL)
Tel 0131.860254 - www.mdsrl.it

Netwaves ..... **pag. 111**
Via Cadorna, 27/31 - 20032 Cormano (MI)
Tel 02.66504794 - www.netwaves.it

Piacenza Expo ..... **pag. 61**
s.s. 10 fraz. Le Mose - 29100 Piacenza
Tel 0523.602711

Pordenone Fiere ..... **pag. 27**
V.le Treviso 1 - 33170 Pordenone
Tel 0434.232111 - www.fierapordenone.it

Precma ..... **pag. 51-57**
Via Fontanino 4 - 23871 Lomagna (LC) - IT
Tel 039.5300590 - www.precma.it

Scuola Radio Elettra ..... **IV cop.**
Via Biturgense, 104 - 00185 Cerbara di Città di Castello (PG)
Tel 075.862911 - www.scuolaradioelettra.it

S.V.M. Elettronica ..... **pag. 87**
Via Sempione, 24 - 21057 Olgiate Olona (VA)
Tel 0331.640569 - www.svmelettronica.com

Studio Fulcro ..... **pag. 33**
Piazzale Kennedy, 1 - 16129 Genova
Tel 010.561111 - www.studio-fulcro.it

Telecontrolli ..... **pag. 97**
Via Valla, 16 - 20141 Milano
Tel 02.84742360 - www.telecontrolli.com

Wireless ..... **pag. 105**
Via Monterosa, 11 - 20049 Milano
Tel 02.48100306 - www.wirelessforum.it

Sommario

245
Novembre
2005

Pratica

La tecnica DDS: Riproduttore di suonerie — 28
di Salvatore Torrisi

Mikrobasic per PICmicro (settima parte): Pilotiamo i motori passo-passo — 48
di Giovanni Di Maria

Il progetto LIRC: Realizzazione di un'interfaccia IR-PC — 64
di Enrico Raffone

CPLD by Example (seconda parte): Utilizzare le CPLD XC9500 e l'ambiente di sviluppo Xilinx — 80
di Agostino Rolando

Usare PROTEUS (seconda parte): ISIS: creazione dei componenti ed editing avanzato — 108
di Maurizio Del Corso

pag. 48

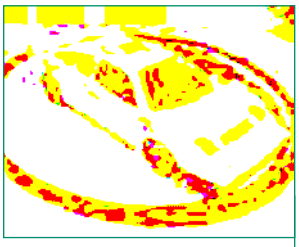
Teoria

Elettronicando (decima parte): Il timer 555 — 14
di Riccardo Nicoletti

L'amplificatore operazionale dalla A alla Z (ottava parte): L'amplificatore differenziale — 76
di Nico Grilloni

Alimentatori switching (undicesima parte): Analisi di uno switching off-line (parte I) — 98
di Romano Bernarducci

Vitamina C (ventitreesima parte): Gestione delle interruzioni — 118
di Antonio Di Stefano



pag. 64

Risorse

Mailbox — 6

Prima pagina — 8

Notepad — 12

Gli appuntamenti di Novembre 2005 — 62

Riflettori su...

Farnell InOne: Tutta l'elettronica in sole 24 ore — 126

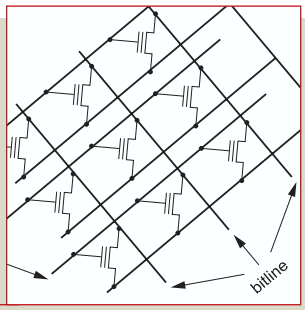
Numero Speciale: 100 idee di progetto per 1000 applicazioni — 128

pag. 98

Speciale

Introduzione alle memorie Flash
di Teodoro Fristachi

pag. 34



Scrivete a:
MAILBOX
REDAZIONE DI
FARE ELETTRONICA
Inware s.r.l.
Via Cadorna, 27/31
20032, Cormano (MI)

Oppure inviate un'email a:
mailbox@farelettronica.com

Mailbox

Questa rubrica ospita le richieste più interessanti pervenute dai lettori.
Per quanto possibile verrà data risposta a tutte le richieste pervenute via email.

TOUCH SWITCH

Vorrei realizzare un circuito per accendere una lampadina a 12V senza impiegare un interruttore meccanico, ma un sensore a sfioramento. Avete una soluzione al mio problema?

Edoardo Livini

Per soddisfare la richiesta del nostro lettore, ecco un classico circuito Touch Switch (figura 1) che impiega una piccola piastra metallica come sensore a sfioramento.
Toccando la piastra, il rumore a 50Hz presente sul dito provoca la conduzione del JFET T1 che fa a sua volta innescare gli stadi successivi provocando l'accensione della lampada.
La lampada potrà essere sostituita con un relè per pilotare carichi maggiori (impiegando un relè sarà necessario inserire un diodo di libera circolazione in parallelo alla bobina del relè con il catodo connesso al positivo di alimentazione).
Il transistor MPF102 potrà anche essere sostituito con un NTE451 o con n ECG451, il 2N3565 con un 2N2222, un BC107, BC108, BC109, NTE123A o con un ECG123A. Il TIP31 può essere rimpiazzato da un ECG196 o un NTE196.

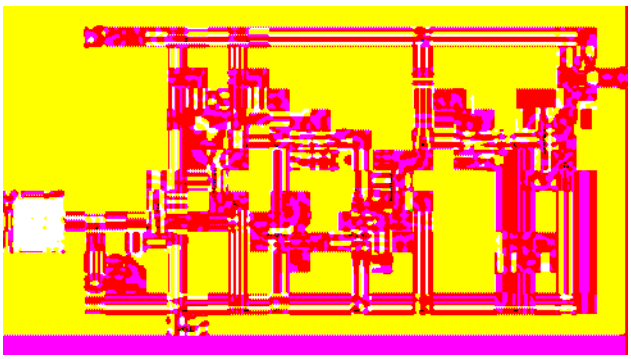


Figura 1 Touch Switch

PORTE TTL DA 400mA

Sul quesito "Le so tutte!" proposto nel numero di settembre riguardante l'interfacciamento di una porta TTL standard con una TTL-LS, ho notato dei valori sospetti nella tabella riportata.
Nella riga relativa alla porta TTL standard si legge I_{ILmax}=400mA ed I_{OHmax}=400mA e tali valori mi sembrano eccessivi. Potrei avere dei chiarimenti in merito?

Fabio Baggiani

L'osservazione del lettore è del tutto lecita. I valori delle due correnti non sono infatti 400mA bensì 400 microAmpere.
Ci scusiamo con i lettori per l'errore.
Con i valori "errati" il fan-out risulta pari a 5, mentre utilizzando i valori reali il fan-out sarebbe risultato pari a 20.
Entrambe le risposte sono state comunque considerate corrette.

PALPATORE

Vorrei innanzitutto complimentarvi con voi per l'articolo "Economica CNC a 3 assi" pubblicato sul numero 243 di Fare Elettronica. Leggendo attentamente l'articolo ho notato che più volte si parla del "palpatore".
Ma cosa è esattamente un "palpatore" e quale è la sua funzione in una macchina a controllo numerico?

Stefano Macchia

“Richieste, chiarimenti, dubbi e commenti dai lettori”

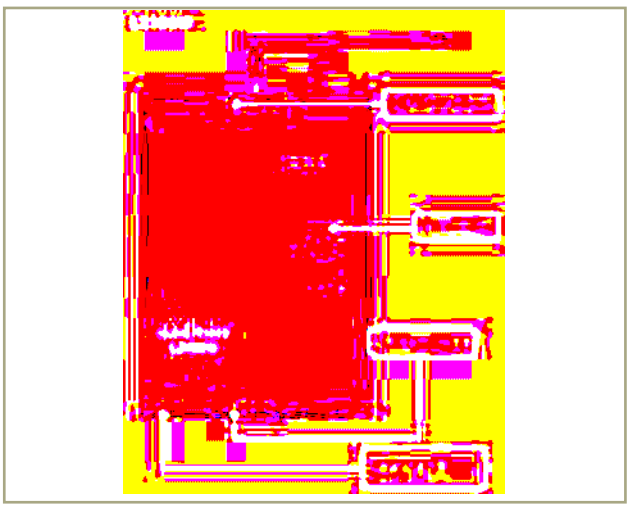
Il palpatore è un pulsante che viene montato in modo fisso sul piano di lavoro (in un angolo che non disturbi troppo), rispetto alla posizione di zero della CNC ed avrà quindi una posizione di coordinate X, Y e Z fisse e ben misurabili.
Poiché alcune lavorazioni prevedono l'utilizzo di più utensili (ad esempio per sgrossare e per rifinire) è ovviamente necessaria l’operazione di cambio di utensile.
Una volta cambiato l’utensile è necessario ricalibrare “l'altezza” della fresa in quanto è praticamente impossibile che il nuovo utensile sporga dal mandrino esattamente come il precedente. Questa è la funzione del palpatore: prima di iniziare una lavorazione l’utensile viene portato sulla zona in cui si trova il palpatore quindi viene abbassato fino a portarlo a contatto con il palpatore stesso.
Quest’ultimo, commutando, informa il SW della nuova calibrazione dell'asse Z.
Anche se non vi sono più lavorazioni, è necessario eseguire l’operazione di calibrazione inizialmente per informare il software su come è stato montato l’utensile e, successivamente per mantenere il software informato sullo stato di usura dell’utensile.

Rubrica di notizie e novità dal mondo dell'elettronica.

Prima pagina

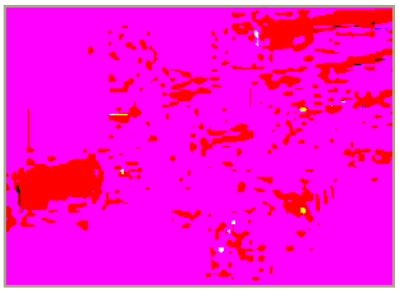
245-01 SENA PERMETTE L'INTERFACCIAMENTO WIRELESS 802.11B DEI DISPOSITIVI SERIALI

Il nuovo **LS100W** è un "Wireless Serial Device Server" e permette il collegamento dei dispositivi seriali RS-232 allo standard delle reti wireless 802.11b. E' nato per collegare dispositivi seriali quali stampanti, bilance, apparecchiature medicali e per la produzione, lettori di bar codes e di tessere magnetiche, POS, e qualsiasi dispositivo di acquisizione tipico dell'infomobilità alle LAN con standard Wireless 802.11b. Dal lato dell'interfaccia seriale, LS100W è dotato di un connettore standard a 9 pin maschio, ingresso per alimentazione 9-30 VDC e supporta velocità di trasferimento dati fino a 115Kbit per secondo.
LS100W supporta sia la modalità "Infrastructure" che quella "Ad-hoc". Nella modalità "infrastructure" l'LS100W trasmette I dati seriali al PC Host mediante un access point. Nella modalità "Ad-hoc" permette il trasferimento dati in full-duplex dei dispositivi seriali attraverso gli apparati LS100W senza l'uso di alcun access point.
LS100W utilizza il protocollo standard 64/128-bit WEP (Wired Equivalent Privacy) per la sicurezza dei dati trasmessi in Wireless.
LS100W supporta i driver COM/TTY per i sistemi operativi Windows e Linux, garantendo il funzionamento dei programmi esistenti e permettendone la facile migrazione verso le nuove tecnologie Wireless LAN. Sono supportate anche entrambe le modalità TCP Server e TCP Client, permettendo al software basato su IP di accedere al dispositivo seriale direttamente conoscendone l'indirizzo IP e il numero della porta TCP.
L'unità e provvista di indicatori LED per l'alimentazione, lo stato, l'attività del link WiFi e dei segnali seriali Rx/Tx. In aggiunta, il Device Server supporta il montaggio a guida DIN oltre al tradizionale "wall mount", garantendo il massimo della flessibilità.



245-02 COMPEL ELECTRONICS AMPLIA LA SUA GAMMA DI CONNETTORI COASSIALI SERIE "N"

Compel Electronics SpA ha incrementato la tipologia dei suoi connettori coax. serie "N" inserendo ulteriori versioni adatte ad essere montate sui nuovi cavi coax. flessibili LMR (Times) ed EC (Eupen). Questi cavi sono alternativi ai tradizionali tipi corrugati, avendo però il vantaggio di una migliore flessibilità di piegatura e di una maggiore velocità e praticità nel montaggio dei connettori. Rispetto ai cavi flessibili RG il vantaggio è quello di avere una minore perdita d'inserzione e una migliore schermatura.
I nuovi connettori coax. "N" ben si adattano alle diverse versioni di questi cavi presenti sul mercato. La gamma è ben coperta da connettori maschi e femmine, e da tipi diritti e ad angolo. Per il montaggio al cavo il collegamento del contatto centrale avviene mediante crimpatura, saldatura o ad innesto (dipende dal modello), mentre lo schermo del cavo viene bloccato mediante crimpatura.



PER SAPERNE DI PIÙ

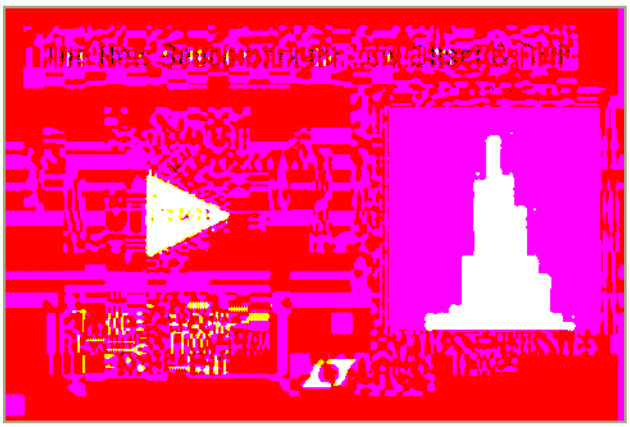
Per approfondire le notizie riportate in questa rubrica, visitate il sito **www.farelettronica.com/primapagina** e seguite le istruzioni.
In alternativa potete scrivere a:

Inware Edizioni
Servizio Prima Pagina
Via Cadorna 27 - 20032 Cormano (MI)

Indicando il codice riportato accanto al titolo della notizia (esempio 245-01).

Tale metodologia di montaggio risulta essere la più veloce ed affidabile. L'impiego di questi connettori è abbastanza esteso e può comprendere cablaggi di apparecchiature interne, base station, code d'antenna, telefonia mobile.

245-03 AMPLIFICATORE CMOS DA 50 µA



Linear Technology Corporation annuncia una nuova famiglia di amplificatori CMOS in grado di offrire una precisione straordinaria in CC, con la corrente di alimentazione più bassa possibile. Con caratteristiche di ingresso CC simili ai migliori amplificatori bipolari di precisione, gli amplificatori operazionali duali LTC6078 e quadrupli LTC6079 offrono una combinazione rivoluzionaria di specifiche che nessun altro amplificatore CMOS presente sul mercato è in grado di eguagliare. Grazie a un'architettura esclusiva, gli amplificatori vantano una minuscola tensione di offset in ingresso di 25 µV Max e una deriva della tensione di offset garantita di 0,7 µV/°C max, assorbendo solo 55 µA di corrente a riposo per canale. Ciò significa una riduzione del consumo di potenza di dieci volte, rispetto agli attuali amplificatori operazionali con simile precisione CC.
Con uno stadio di ingresso rail-to-rail, gli LTC6078/LTC6079 mantengono la loro eccezionale precisione di ingresso sull'intera gamma di modalità comuni. La bassissima corrente di polarizzazione, 1 pA max, consente un'ottima precisione CC con elevata impedenza sorgente e resistori di retroazione. Inoltre, l'eccellente CMRR di 110 dB e l'elevato guadagno di tensione di 120 dB assicurano prestazioni di precisione. Operando su una gamma di alimentazione da 2,7 V a 5,5 V, le uscite degli amplificatori hanno un'escursione "rail-to-rail". La corrente di alimentazione si riduce a meno di 2 uA, grazie alla funzione di arresto.
L'LTC6078 dual è fornito in un package MSOP a 8 pin e in un minuscolo package DFN 3x3mm. L'LTC6079 quad è disponibile nel package SSOP a 16 pin e DFN 5x3mm. Con piena garanzia di funzionamento nei range di temperatura per applicazioni commerciali, industriali e automobilistiche.

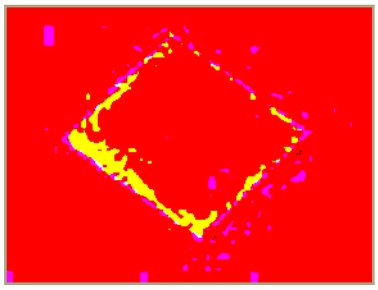
245-04 NUOVI IC DI POWER MANAGEMENT MICROCHIP COMBINANTI FUNZIONI DISCRETE

Microchip introduce i nuovi IC di power management TC1303/TC1304 e TC1313. I dispositivi combinano – all'interno di una soluzione single-chip - un regolatore switching buck sincrono, un regolatore a bassa caduta (LDO) e una funzione di power-good. Tutte e cinque le opzioni integrano il regolatore buck da 500 milliampere (mA) e un LDO da 300mA.
Questi dispositivi garantiscono consumi contenuti, costi ridotti e un ingombro minimo su scheda, mettendo a disposizione una soluzione ideale per un ampio spettro di sistemi alimentati a batteria o con doppia tensione di uscita. Il regolatore buck sincrono è un convertitore DC-DC switching estremamente efficiente. Nelle condizioni di carico più pesanti, il converter opera in modalità PWM a una frequenza di switching di 2.0MHz.
In condizioni di carico leggero, per esempio in

modalità standby, il converter commuta in modalità PFM (pulse frequency modulation) riducendo così i consumi. Il regolatore prevede opzioni con tensioni di uscita regolabili o fisse. Il regolatore lineare LDO a basso rumore supporta carichi fino a 300mA garantendo una caduta di tensione estremamente bassa (137mV a 200mA typ) ed è stabile con un solo condensatore ceramico da 1.0uF sull'uscita.
Tutti i dispositivi offrono varie dotazioni di protezione tra le quali under-voltage lockout (UVLO), sovratemperatura e corto circuito sull'uscita.
I dispositivi TC1303/TC1313 offrono controlli individuali di shutdown per i due regolatori, mentre il dispositivo TC1304 utilizza la funzione power-good insieme al pin SHDN per generare una funzione di power-sequencing.
Tale funzione prevede che durante il power-up l'LDO arrivi a operare all'interno delle specifiche prima di attivare il regolatore: durante il power-down invece, l'uscita del regolatore buck viene disabilitata prima dell'uscita dell'LDO. Per dimostrare l'efficacia di queste soluzioni di power management, Microchip offre la scheda demo TC1303BDM-DDBK1.

245-05 SENSORE DI IMMAGINE CMOS DA 3 MEGAPIXEL PER TELEFONINI CON FOTOCAMERA INTEGRATA

Cypress Semiconductor ha annunciato il proprio ingresso nel mercato dei telefonini con fotocamera integrata grazie alla realizzazione di un sensore di immagine in tecnologia CMOS da 3 Megapixel.



Caratterizzato da un costo particolarmente competitivo, il nuovo sensore CYIWOSC3000A garantisce un livello qualitativo del tutto assimilabile a quello delle fotocamere digitali.
Il sensore permette di ottenere immagini a elevata risoluzione utilizzando il formato ottico da 1/3", molto diffuso nei dispositivi a più bassa risoluzione, in modo da consentire ai costruttori di procedere all'aggiornamento dei loro dispositivi senza dover affrontare le spese legate all'integrazione di nuove lenti ottiche.
Il nuovo dispositivo sfrutta la tecnologia sviluppata da Cypress Semiconductor che prevede l'uso di pixel realizzati mediante tre transistor (3T), grazie alla quale è possibile migliorare la sensibilità in condizioni di bassa luminosità senza incrementare il numero di transistor o penalizzare la qualità dell'immagine. Esso inoltre fa ricorso alla tecnologia brevettata Autobrite™ che, garantendo un range dinamico molto elevato, permette di distinguere in maniera netta le aree buie e luminose di una stessa immagine.
Il visualizzatore di immagini CYIWOSC3000A per telefonini dotati di telecamera è un sensore in formato ottico da 1/3" con una matrice di pixel attiva formata da 2048 x 1536 pixel. La tecnologia Autobrite permette al sensore di operare su un range dinamico fino a 72 dB, garantendo l'acquisizione di ogni dettaglio all'interno di una scena.
Caratterizzato da una velocità di 30 fotogrammi al secondo (fps – frame per second) alla massima risoluzione e di 83 fps a una risoluzione di 640 x 480 pixel, il nuovo sensore è dotato di un sistema ERS – Electronic Rolling Shutter – e dispone a bordo di un convertitore A/D a 12 bit.
Il sensore integra una serie di controlli programmabili – compresi dimensioni del frame, velocità dell'immagine, esposizione, panning elettronico, binning di righe e colonne, immagine lineare e non, correzione dell'offset del nero, flip e mirroring – in un unico chip.

Presenti e assenti per la meteorologia

STAZIONE METEOROLOGICA

STAZIONE METEOROLOGICA

PROBE STRUMENTI

TAVOLA DISCUTERE

VARIE

Dal blocco note di Fare Elettronica una raccolta di idee da tenere sempre a portata di mano.

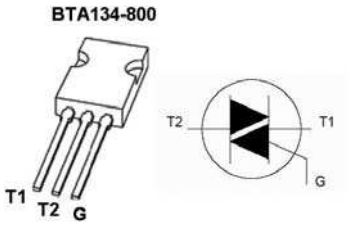
Notepad

Questa rubrica ha lo scopo di fornire degli schemi applicativi o idee di progetto dei componenti elettronici più interessanti, selezionati per voi dalla redazione. Tutti gli schemi presentati sono elaborazioni di quelli ufficiali proposti dai produttori nella documentazione ufficiale.

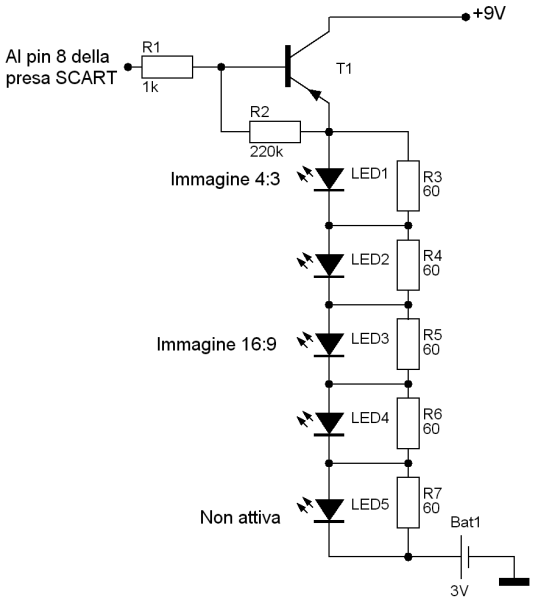
MODELLO SPICE PER TRIAC BTA134

Per poter simulare un triac utilizzando il simulatore PSPICE o il simulatore incluso nel pacchetto Proteus, è necessario disporre del modello opportuno. Di seguito è riportato il modello SPICE per un BTA134-800 con il relativo simbolo circuitale e package.

```
*SRC=BT134-800;BT134-800;TRIACS;PHILIPS;800V 4A
*SYM=TRIAC
.SUBCKT BT134-800  1  2  3
*      TERMINALS: MT2 G MT1
QN1  5 4 3  NOUT OFF
QN2 11 6 7  NOUT OFF
QP1 6 11 3  POUT OFF
QP2  4 5 7  POUT OFF
DF  4  5  DZ OFF
DR  6 11  DZ OFF
RF  4 6  80MEG
RT2 1 7  0.141
RH  7 6  35
RGP 8 3  15.5
RG  2 8  31.7
RS  8 4  34.5
DN  9 2  DIN OFF
RN  9 3  457
GNN  6 7 9 3 16.6M
GNP  4 5 9 3 3.91M
DP 2 10  DIP OFF
RP 10 3  457
GP  7 6 10 3 15.3M
.MODEL DIN  D (IS=115F)
.MODEL DIP  D (IS=115F N=1.5)
.MODEL DZ   D (IS=115F N=1.5 IBV=10U BV=800)
.MODEL POUT PNP (IS=115F BF=5 CJE=508P TF=43.4U)
.MODEL NOUT NPN (IS=115F BF=20 CJE=508P
 CJC=102P TF=2.89U)
.ENDS
```



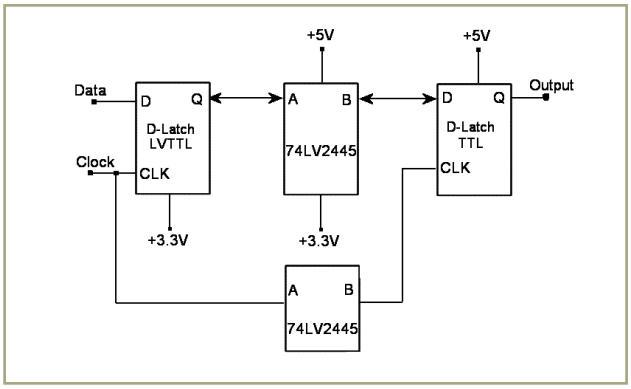
STATO DELLA PRESA SCART



Il pin 8 della presa SCART del TV, VCR, DVD, ricevitori satellitari e terrestri fornisce informazioni sul formato dell'immagine e sull'attività o meno della presa scart stessa. Se la tensione a questo pin è compresa tra 0V e 2V, significa che la presa non è attiva; se la tensione è compresa tra 4,5V e 6V significa che l'immagine è in 16:9; infine se la tensione è compresa tra 9,5V e 12V significa che l'immagine è in 4:3. Il circuito proposto è un comparatore multilivello in grado di accendere un LED in base alla tensione al pin 8 della presa scart.

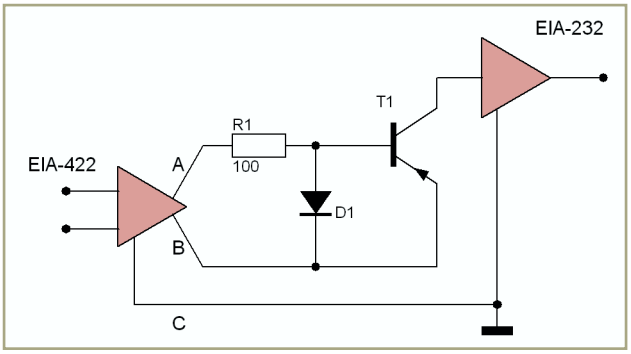
INTERFACCIA LVTTL-TTL

Il circuito di figura consente di interfacciare porte logiche Low-Voltage-TTL (quindi con livelli 0/3,3V) con porte TTL (quindi con livelli 0/5V) e viceversa. Il circuito impiega un traslatore di livello 74LV2445 e l'esempio in figura mostra l'interconnessione tra due D-Latch.



INTERFACCIA EIA-422/EIA-232

Lo standard EIA-422 prevede una comunicazione differenziale su bus a due fili, mentre l'EIA232 trasmette su una singola linea. I dati in 422 sono inviati in logica positiva, mente in 232 i dati vengono trasmessi in logica negativa. Il circuito di figura impiega una resistenza, un diodo ed un transistore bipolare, per operare una semplicissima conversione da EIA-422 verso EIA-232.



Il temporizzatore 555 è uno dei componenti "storici" dell'elettronica, come il famoso 741 di cui ci siamo già interessati. È un circuito integrato ampiamente utilizzato come multivibratore monostabile ed astabile per realizzare circuiti temporizzatori, generatori di impulsi e forme d'onda. Fu introdotto in origine dalla Signetics con tecnologia bipolare, ma adesso viene prodotto da molti Costruttori anche in tecnologia CMOS. Presentiamo la sua struttura interna e simuliamo alcune semplici applicazioni con circuiti multivibratori.

UN COMPONENTE STORICO: IL 555

Contenitore

Il circuito integrato 555 è prodotto da diverse Case, con differenti sigle. Lo si trova in commercio con le denominazioni LM555, NE555, SA555, a seconda del produttore. Può essere contenuto in un contenitore a 8 piedini circolari tipo TO-99 (figura 1), in un mini dip a 8 piedini (figura 2) o in un dip a 14 piedini (figura 3). Nelle figure 1,2 e 3 sono riportate alcune delle piedinature più comuni.

Circuito interno

In figura 4a è riportato lo schema a blocchi interno, mentre in figura 4b lo schema è ripreso dal datasheet Fairchild con nomenclatura inglese. Possiamo osservare che il dispositivo è composto da due comparatori, di cui uno è detto di **trigger** ed al quale fa capo l'ingresso di *trigger* (pin 2); tale comparatore ha come tensione di riferimento (punto B) quella ottenuta dalla par-

Figura 1 *Piedinatura temporizzatore 555 in package TO-99*

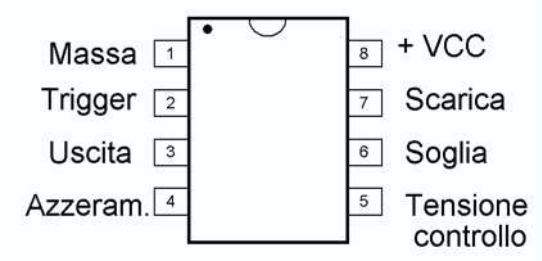


Figura 2 *Piedinatura temporizzatore 555 in package 8 pin*

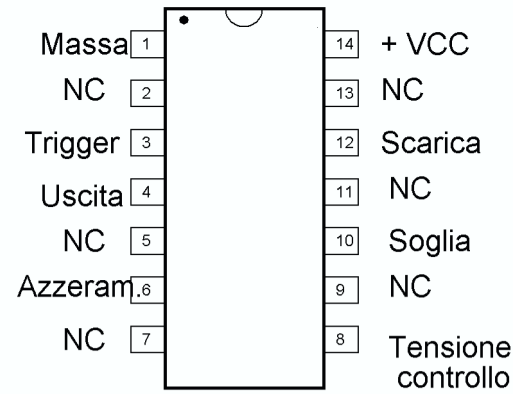
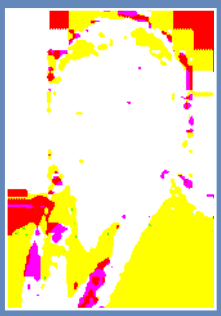


Figura 3 *Piedinatura temporizzatore 555 in DIP 14 pin*

Il timer 555



di Riccardo Nicoletti
(r.nicoletti@farelettronica.com)

tizione di V_{CC} sulle tre resistenze da 5 kΩ (ecco da dove deriva la sigla "555"). L'altro comparatore è detto di **soglia** ed al suo ingresso è collegato appunto il piedino di *soglia* (pin 6) e la cui tensione di riferimento (pin 5, punto A) è pari a $2/3V_{CC}$. L'uscita del comparatore di trigger è collegata all'ingresso S di un flip-flop SR non cadenzato. L'altro ingresso del flip-flop (R) è comandato dall'uscita del comparatore di soglia. Il piedino di *azzeramento* è collegato al *reset* del flip-flop.
L'uscita del flip-flop stesso (Q), pilota un buffer invertente, cioè un dispositivo funzionante da porta not a bassa impedenza di uscita; l'uscita del buffer invertente costituisce l'uscita del timer (pin 3).
Allo stesso tempo l'uscita Q del flip-flop pilota un transistor ad emettitore comune il cui

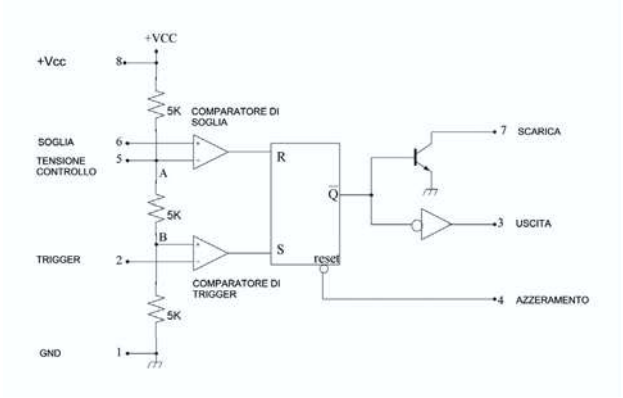


Figura 4a Schema a blocchi interno del temporizzatore 555

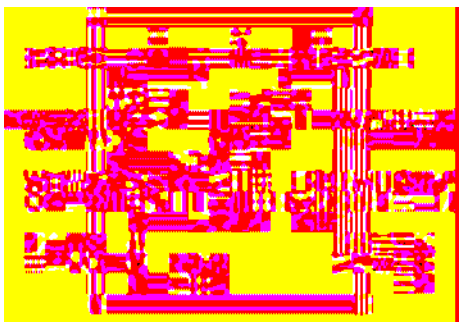


Figura 4b Schema a blocchi interno del temporizzatore 555 (Fairchild)

collettore è collegato con il piedino di scarica del timer. Uno sguardo allo schema a blocchi interno riportato nei datasheet del componente di diverse marche è un utile esercizio per assimilare la struttura interna del componente, e per vedere le eventuali differenze adottate dai singoli Costruttori.

Piedinatura

Con riferimento alla figura 4a, vediamo in dettaglio le funzioni degli otto piedini del circuito.

1. **Massa** - La sua funzione è evidente: si tratta del collegamento a massa (comune) per le tensioni di alimentazioni e di segnale.

2. **Trigger** (attivo basso) - È usato per dare inizio ad un ciclo di temporizzazione. Il timer 555 viene fatto partire abbassando il livello della tensione su questo piedino al di sotto di 1/3 V_{CC}.
   Normalmente questo piedino viene mantenuto ad una tensione maggiore. Il trigger è sensibile al livello, pertanto forme d'onda a variazione lenta (come le onde sinusoidali) possono essere usate, con pari risultato, al posto delle onde di commutazione tradizionali (come le onde rettangolari). In generale, per ottenere un avviamento affidabile, è sufficiente che il segnale di trigger duri almeno un secondo. (Attenzione: nel dip a 14 pin è il pin 3).

3. **Uscita** - Da questo terminale possono essere prelevati gli impulsi di temporizzazione generati dai circuiti interni costituenti il timer. È da notare che l'uscita del temporizzatore è paragonabile ad un segnale di tipo digitale, potrà assumere cioè due soli livelli ben definiti: uno basso (livello "0", low) corrispondente ad una tensione prossima a quella di massa, ed uno alto (livello "1",

high) corrispondente ad una tensione prossima a quella di alimentazione del timer (+V_{CC}). Per questo motivo le uscite del 555 sono facilmente interfacciabili con circuiti digitali di tipo TTL e CMOS (è il piedino 4 nel dip a 14 pin).

4. **Azzeramento** (RESET, attivo basso) - Questo terminale è usato per azzerare il flip-flop interno e riportare quindi l'uscita (piedino 3) a livello basso. Il livello di soglia della tensione di azzeramento (tensione di reset) è di 0,7 V ed è necessario fornire una corrente di 100µA per ottenere l'azzeramento.
L'azzeramento avviene senza tener conto del valore di tensione di alimentazione (+V_{cc}) e senza tener conto dei segnali eventualmente presenti agli altri piedini (è il piedino 6 nel dip a 14 pin).

5. **Tensione di controllo** - permette ad una tensione esterna di controllare i livelli di commutazione dei comparatori interni. Può essere usato per pilotare oscillatori, temporizzatori ed altri dispositivi controllati in tensione (Pin 8 nel dip a 14 pin).

6. **Soglia** (TRESHOLD, attivo alto) - È uno degli ingressi del comparatore di soglia. Viene utilizzato per l'azzeramento dell'uscita, ottenuto elevando la tensione, ad esso applicata, ad un livello di 2/3V_{CC}. La funzione di questo

Caratteristiche generali	Condizioni di prova	Min.	Typ.	Max
Campo della tensione di alimentazione		4.5V		18V
Corrente di alimentazione	V_{CC}=5V; R_u=∞		3mA	5mA
(stato basso)	V_{CC}=15V; R_u=∞		10mA	12mA
(stato alto)			<1mA	
Errore periodo (configurazione astabile)	R_1; R_2=(1-100) KΩ			
	C =0,1µF		1.5%	
Deriva con la temperatura (periodo)	V_{CC}=15V		90 ppm/°C	
Deriva con V_{CC} (periodo)			0.15%/V	
Livello di tensione di controllo	V_{CC}=15V	9.6V	10V	10.4V
	V_{CC}=5V	2.9V	3.33V	3.8V
Tensione di soglia	V_{CC}=15V	9.4V	10V	10.6V
	V_{CC}=5V	2.7V	3.33V	4V
Tensione di trigger	V_{CC}=15V	4.8V	5V	5.2V
	V_{CC}=5V	1.45V	1.67V	1.9V
Corrente di soglia			0.1µA	0.25µA
Corrente di trigger			0.5µA	0.9µA
Tensione di uscita (livello basso)	V_{CC}=15V			
	I_{sink}=10mA		0.1V	0.15V
	I_{sink}=100mA		2V	2.22V
Tensione di uscita (livello alto)	V_{CC}=5V			
	I_u=100mA	13V	13.3V	
Tempo di turn off	V_{reset}=V_{CC}		0.5µs	2µs
Tempo di salita (dell'uscita)			100ns	200ns
Tempo di discesa (dell'uscita)			100ns	200ns

Tabella 1 *Caratteristiche del timer 555*

Reggio

la fiera dell'elettronica
& del radioamatore

Con oltre 150 espositori
e tanta informatica ai prezzi
più bassi d'Italia

MOSTRA MERCATO

19-20

NOVEMBRE 2005

Domenica 20 novembre
09.00 - 18.00

prevendita biglietti dalle ore 08.00

Biglietto intero € 6,00
Ridotto € 4,00 scolaresche
Parcheggio gratuito

- COMPUTER
- ELETTRONICA
- RADIANTISMO
- EDITORIA
- TELEFONIA
- TV-SATELLITARE
- HOBBISTICA

Organizzazione:
COMPUTER & RADIO
Centro Fiere Reggio Emilia
Via Caprera n. 1 - 42124 Reggio Emilia
tel. 0522 - 451411 - fax 0522 - 451414

piedino dipende dal livello, per cui possono essere applicate anche forme d'onda a variazione lenta (piedino 10 nel dip a 14 pin).

7. **Scarica** (DISCHARGE) - È collegato al collettore di un transistor (transistor di scarica interno) che è interdetto quando l'uscita (pin 3) è alta, mentre è in saturazione quando l'uscita è bassa. Questo interruttore a transistor permette un "ancoraggio" a massa degli opportuni nodi della rete di temporizzazione (piedino 12 nel dip a 14 pin).

8. **+Vcc** - È il terminale di ingresso della tensione positiva di alimentazione. Il timer funziona bene con tensioni di alimentazione che vanno da +4.5 V a +15 V in corrente continua (piedino 14 nel dip a 14 pin).

Caratteristiche elettriche

Le caratteristiche del timer 555 sono riassunte nella tabella 1.

Esistono in commercio dei timer integrati realizzati con porte CMOS (come ad esempio il TS555 della STM, il tipo 7555 prodotto dalla Intersil, o il TLC551 della Texas Instruments).
I vantaggi del timer a CMOS sono vari:

- Il consumo è assai inferiore rispetto al tipo tradizionale.
- Può funzionare con basse tensioni di alimentazione (anche 1V).
- Ha dentro di se un numero minore di transistori di commutazioni (detti anche "spiker").
- Ha un comportamento migliore alle alte frequenze.

IL 555 COME OSCILLATORE

Un uso classico del 555 prevede il suo impiego per la realizzazione di un *oscillatore a rilassamento*. Gli "**oscillatori a rilassamento**" sono circuiti molto semplici e basano la loro oscillazione sulla carica e scarica di un condensatore attraverso un resistore. Questa tipologia di oscillatori è particolarmente economica e semplice, e con la dovuta attenzione nel progetto presenta anche una discreta stabilità in frequenza.

Vediamo un'applicazione del 555 in tal senso. Lo schema è riportato in figura 5a.
L'uscita va alta (cioè con valore molto vicino a $+V_{cc}$) quando viene posto un segnale di TRIGGER mantenendolo fino a quando il pin SOGLIA supera il valore $+2/3V_{cc}$; a questo punto l'uscita va bassa (vicino a 0V) e il transistor di SCARICA è portato in conduzione. L'ingresso TRIGGER è attivo quando viene portato sotto $1/3V_{cc}$ e la SOGLIA è attivata sopra $2/3V_{cc}$.

Nota: *nella spiegazione del funzionamento interno del componente si fa riferimento genericamente all'uscita Q del flip-flop. Se si tratti di Q o, nello specifico, del suo complemento $\overline{Q}$ sarà chiaro dal contesto.*

Il modo più semplice di comprendere questo funzionamento è di vedere un esempio.

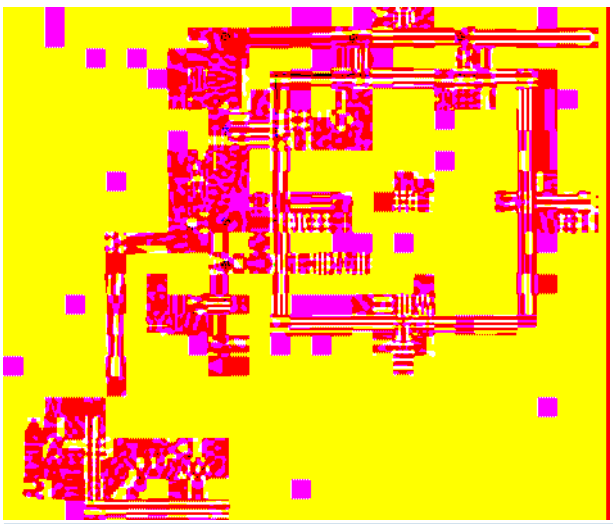


Figura 5a Il 555 collegato come oscillatore (Fonte: [1])

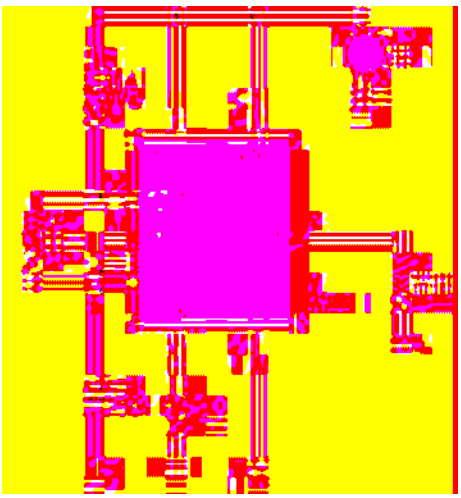


Figura 5b Schema del 555 collegato come oscillatore per CadLogix

Riferiamoci alla figura 5a. Il 555 è collegato esternamente ad una rete RC formata dalla due resistenze R_A ed R_B con il condensatore C. Supponiamo il condensatore inizialmente scarico. Quando applichiamo l'alimentazione l'uscita va ALTA poiché il pin TRIGGER è applicato al condensatore che è scarico. Il pin 7 di SCARICA è ad un potenziale basso quindi manda in interdizione il BJT di scarica. Il condensatore comincia a caricarsi verso V_{cc}=10V tramite la serie R_A+R_B.
Quando il suo potenziale raggiunge 2/3 V_{cc} il pin SOGLIA manda l'uscita BASSA e il pin SCARICA accende il bjt; il condensatore C si scarica attraverso R_B mediante la conduzione del BJT stesso.
Il tutto si ripete da questo momento, con il potenziale ai capi di C che va da 1/3V_{cc} a 2/3V_{cc} e con periodo

$$T = 0.693\ (R_A + 2R_B)\ C.$$

L'uscita che normalmente viene utilizzata è l'onda quadra sul pin di uscita 3, riportata in figura 6. In figura 5b è mostrato lo schema utilizzato per la simulazione; si noti il resistore collegato al pin 5, necessario unicamente perché il simulatore non accetta pin non connessi.
Ci preme evidenziare un dettaglio sul funzionamento di questo componente. Le tre resistenze interne da 5 kΩ producono dei potenziali di riferimento sul morsetto "+" del comparatore 2 e sul "-" del comparatore 1; la circuiteria esterna che viene aggiunta serve a realizzare *fenomeni di carica e scarica* che, in rapporto ai potenziali fissati sui comparatori, provocano eventuali commutazioni. L'aggiunta di un diodo complicherà le cose in modo molto interessante nelle ultime due configurazioni.
Il 555 è un oscillatore di tutto rispetto, con una stabilità vicina al 1%. Mantiene una buona stabilità in frequenza anche in presenza di fluttuazioni della tensione di alimentazione, che vengono eliminate grazie alla funzione di soglia presente.

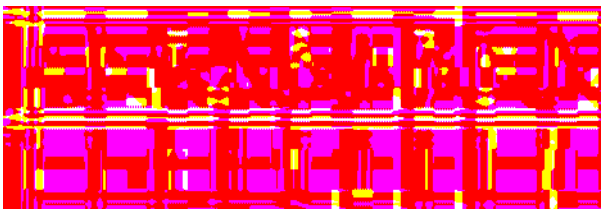


Figura 6 *Simulazione del 555 come oscillatore (R_A=10KΩ, R_B=10KΩ, C=0.1μF, T=2ms)*

TIMER 555 (CONFIGURAZIONE ASTABILE)

Vediamo alcuni oscillatori a rilassamento nelle loro configurazioni tipiche. L'esempio mostrato in precedenza prende il nome di "*multivibratore astabile*". Come si comprende dal nome, in questa configurazione non esiste uno stato "stabile", quindi l'uscita è una forma d'onda che commuta continuamente: il circuito è appunto "*astabile*". Lo schema elettrico del temporizzatore astabile di figura 5a viene riportato in figura 7.

Supponendo il condensatore C inizialmente scarico all'istante t=0, l'uscita del comparatore superiore è bassa, mentre quella del comparatore inferiore è alta; avremo dunque R=O e S=1, di conseguenza Q=O. L'uscita (piedino 3) è alta (+V_{CC}) e il transistor è interdetto. In queste condizioni il condensatore inizierà a caricarsi e la tensione ai suoi capi, applicata ai due compara-

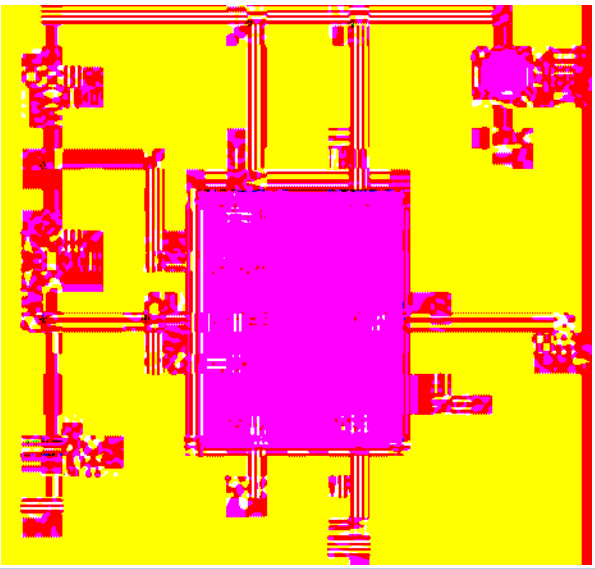


Figura 7 *Schema elettrico del timer in configurazione di oscillatore astabile*

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	?

Figura 8 *Tavola della verità per il flip-flop RS interno*

tori, tenderà a raggiungere la tensione V_{CC} con costante tempo:

$$\tau_1 = (R_A + R_B) \cdot C$$

Raggiunto il valore di 1/3 V_{cc} il comparatore di trigger cambia stato (S passa a 0) e il flip-flop, avendo agli ingressi R=0 e S=0 conserva lo stato precedente (si veda la figura 8 con la tavola della verità). Il condensatore continuerà cosi a caricarsi fino a raggiungere il valore di 2/3 V_{cc}. In questo istante anche l'uscita del comparatore di soglia cambia stato e R passa a 1. A questo punto il flip-flop, avendo in ingresso S=0 ed R=1, si resetterà e $\overline{Q}$ passerà a 1. Il tempo impiegato dalla tensione sul condensatore C per andare dal valore zero al valore 2/3 V_{cc} vale (omettendo i conti):

$$T_0 = \tau_1 \ln 3$$

dove ln3 = 1.098 è il logaritmo naturale di 3.
Essendo Q=1 l'uscita commuterà allo stato basso mentre il transistor si porterà alla saturazione. In queste condizioni il condensatore inizierà a scaricarsi tramite R_B e il transistor con costante di tempo:

$$\tau_2 = R_B \cdot C$$

Quando la tensione sul condensatore raggiunge il valore di 1/3 V_{cc} l'uscita del comparatore di trigger cambia stato ed S passa a 1. Essendo già R=0 il flip-flop si setta e l'uscita va bassa.
Il transistor passerà nuovamente all'interdizione e l'uscita si porterà a livello alto.
Il tempo che la tensione ai capi del condensatore impiega per portarsi da 2/3 V_{cc} ad 1/3 V_{cc} vale:

$$T_1 = \tau_2 \ln 2$$

Essendo il transistor interdetto, e l'unica uscita disponibile è a valore alto, il condensatore inizierà nuovamente a caricarsi con costante tempo:

$$\tau_1 = (R_A + R_B) \cdot C$$

Quando la tensione ai suoi capi raggiunge il valore di 2/3 V_{cc} l'uscita del comparatore di soglia commuta al valore alto (R=1) ed essendo S=0 il flip-flop si azzera forzando l'uscita al valore 1. Il transistor torna di nuovo in saturazione provocando la scarica del condensatore attraverso R_B riprendendo il ciclo. Il tempo T_2 che la tensione sul condensatore impiega per andare da 1/3 V_{cc} a 2/3 V_{cc} vale:

$$T_2 = \tau_1 \ln 2$$

quindi in definitiva

$$T_1 = R_B \cdot C \cdot \ln 2$$
$$T_2 = (R_A + R_B) \cdot C \cdot \ln 2$$

Come possiamo osservare i semiperiodi T_1 e T_2 dell'onda quadra di uscita non sono uguali (figure 9 e 10). Se vogliamo $T_1 \cong T_2$ occorre che R_A sia trascurabile nella somma con R_B, e quindi è necessario imporre la condizione:

$$R_B >> R_A$$

Nel datasheet è inoltre mostrato un diagramma che fornisce la frequenza dell'oscillazione di uscita in funzione dei tre valori dei componenti R_A, R_B e C.

TIMER 555 (CONFIGURAZIONE MONOSTABILE NON RETRIGGERABILE)

Il circuito "monostabile" possiede uno stato stabile. Lo schema è quello di figura 11, mentre in figura 12 è mostrato lo schema interno completo. Il circuito è un po'complesso, poiché la presenza di un diodo rende tutto (apparentemente) più difficile da comprendere; vediamo di spiegarne il funzionamento nella maniera più chiara possibile.

Nota: *il simbolo V_{2+} denota l'ingresso "+" del comparatore 2; il simbolo V_{1-} si riferisce all'ingresso "-" del comparatore 1.*

All'accensione l'ingresso V_{IN} sia aperto. Il diodo è interdetto e l'ingresso V- del comparatore di trigger è alla tensione $+V_{CC}$ tramite R_1. Poiché V_{2+} è a tensione $V_{CC}/3$, l'uscita del comparatore 2, cioè l'ingresso S del flip-flop, è al valore 0. Sempre all'accensione il condensatore C, inizialmente scarico, si comporta come un corto cir-

Grande FIERA ELETTRONICA

FORLÌ 3-4 dicembre 2016

Fiera di Forlì - via Punta di Ferro

the Best!

Fiera Astronomia

Photo Cine Video
2 Sale Rosa con modelli

RADIO EXPO'
MERCATINO delle
RADIOCOMUNICAZIONI

DISCHI e CD
USATI e DA
COLLEZIONE

CONVEGNI
ESPOSIZIONI
ELETTRICO e
ELETTRONICO

NEWS ON LINE!

cuito, portando così a potenziale di massa l'ingresso V_{1+}. Poiché V_{1-} è alla tensione di 2/3 V_{cc}, anche l'uscita del comparatore 1, cioè l'ingresso R del flip-flop, è al valore 0.
Il flip-flop, avendo entrambi gli ingressi a zero, manterrà l'uscita Q al valore assunto al momento dell'accensione (si veda la figura 8).
Supponiamo Q=0: il bjt è interdetto e il condensatore C tende a caricarsi al valore $+V_{CC}$ attraverso la resistenza R seguendo la legge:

$$V_{c(t)} = V_{cc}\left[1 - e^{\frac{t}{\tau}}\right]$$

con costante di tempo

$$\tau = RC$$

Quando la tensione sul condensatore raggiunge il valore 2/3 V_{cc} l'uscita del comparatore 1 passa al valore 1. Questo avviene in un tempo:

$$t_0 = \tau \ln 3$$

Ora il flip-flop ha l'ingresso S=0 e l'ingresso R=1; l'uscita Q si porterà nello stato 1. Il bjt saturerà e la V_{CE} si porterà al valore di circa 0.2 V. Il condensatore C si scaricherà attraverso la corrente di saturazione del BJT, quindi molto rapidamente. L'uscita del comparatore 1 si porterà al valore 0 non appena la tensione sul condensatore sarà scesa al di sotto del valore di 2/3 V_{cc}.
Gli ingressi R e S del flip-flop sono ora entrambi a zero e l'uscita Q rimane nello stato 1; l'uscita V_{OUT} permane quindi nello stato 0. Questa è una *condizione stabile*.
Se viceversa supponiamo Q inizialmente nello stato 1 il BJT sarà saturo e il condensatore si caricherà velocemente alla tensione V_{CEsat}. V_{1-} sarà sempre maggiore di V_{1+} e l'uscita del comparatore 1 rimarrà permanentemente a zero quindi Q sarà sempre a 1 e l'uscita V_{OUT} sarà sempre a zero. Abbiamo quindi dimostrato che con il circuito di comando aperto lo stato stabile sarà sempre comunque il seguente:

$$\left.\begin{array}{lll} V_{2-} = +V_{cc} & V_{2+} = V_{cc}/3 & S = 0 \\ V_{1-} = \frac{2}{3}\cdot V_{cc} & V_{2+} = 0{,}2[V] & R = 0 \end{array}\right\} \begin{array}{l} Q=1; \\ BJT=saturo; \\ V_{OUT}=0 \end{array}$$

Introduciamo ora il segnale di comando V_{IN} riportato in figura considerando anche C' inizialmente scarico, nell'istante t=0 il catodo del diodo viene portato istantaneamente al potenziale di massa e quindi il diodo entra in conduzione. Il potenziale V_{2-} si porta quindi al valore della sua tensione di conduzione V_γ.
Se 1/3 $V_{cc} > V_\gamma$ l'uscita del comparatore 2 (S) si porta ad 1. Poiché R=0, l'uscita Q del flip-flop passa nello stato 0 ed il BJT si interdice; la V_{CE} segue la tensione sul condensatore che tende al valore V_{CC} con legge:

$$V_c(t) = V_{cc}\,(V_{CEsat} - V_{cc})\,e^{\frac{t}{\tau}}$$

dove $\tau = RC$.

L'uscita del comparatore 1 rimane a 0 finché $V_c = V_{1+}$ rimane al di sotto di 2/3 V_{cc}. Ciò avviene per un tempo pari a:

$$t'_0 = \tau \cdot \ln \frac{3\cdot(V_{cc} - V_{CEsat})}{V_{cc}}$$

Assumendo $V_{CEsat} = 0$ si ha:

$$t'_0 = t_0 = \tau \ln 3$$

Trascorso il tempo t_0 l'uscita del comparatore 1

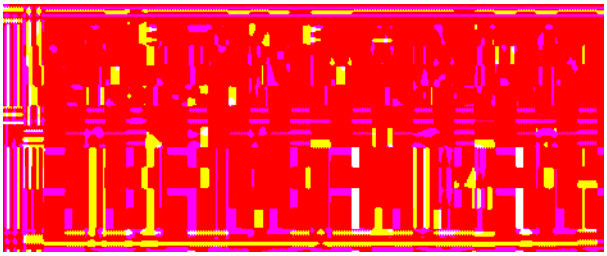


Figura 9 *Forme d'onda ottenute dalla simulazione con CadLogix (R_A=1 KΩ, R_B=10KΩ, C=0.1μF)*

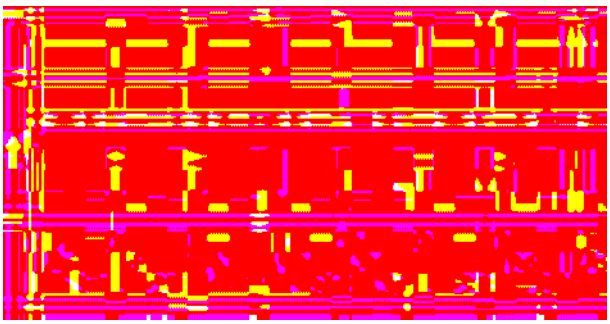


Figura 10 *Forme d'onda ottenute dalla simulazione con CadLogix si notino le tensioni ai capi di R_A ed R_B (seconda e terza forma d'onda)*

passa a 1 (quindi R=1). Possiamo ora vedere l'andamento delle tensioni che coinvolgono il comparatore 2 ed il circuito di comando. Abbiamo visto che all'istante t=0 del segnale di comando, il condensatore, comportandosi come un corto-circuito fa si che il diodo entri in conduzione e porti l'uscita del comparatore 2 (S) allo stato 1.
A regime C' tenderà a comportarsi come un circuito aperto e il diodo tornerà a polarizzarsi inversamente. Il condensatore C' tende a caricarsi alla tensione V_{CC} con una costante tempo dell'ordine di $C' \cdot (R_1 // R_2)$ ed il diodo resterà in conduzione finché $V_{c'}$ non raggiunge il valore di

$$V_{CC} - V_\gamma$$

Essendo

$$V_{c'(t)} = V_{cc}\left(1 - e^{-\frac{t}{\tau'}}\right)$$

il tempo t' che la $V_{c'}$ impiega a raggiungere il valore di $V_{CC} - V_\gamma$ è dato da:

$$t' = \tau' \cdot \ln \frac{V_{cc}}{V_\gamma}$$

Trascorso questo tempo, che può essere reso molto breve scegliendo opportunamente τ', il diodo torna ad interdirsi e la tensione V_{2-}, torna al valore V_{CC} e l'uscita del comparatore 2 (S) torna al valore zero. Quindi se si sceglie opportunamente $t' << t'_0$ siamo sicuri che nell'istante t_0, nel quale l'uscita nel comparatore 1 (R) passa a 1, l'uscita del comparatore 2 (S) era già a zero da un tempo pari a $t'_0 - t'$. Nell'intervallo di tempo compreso tra t'e t'_0 gli ingressi del flip-flop sono entrambi a zero quindi l'uscita Q non varia, resta cioè a zero.
Nell'istante t'_0 nel quale l'ingresso R del flip-flop passa a 1, l'uscita Q passa a 1 (ricordiamo che $S(t_0)=0$). Il BJT satura a partire dalla regione attiva ed il condensatore C si scarica velocemente con corrente costante. Non appena la tensione

sul condensatore C scende al di sotto di 2/3 V_{cc} l'uscita del comparatore 1 (R) si porta allo stato zero. Siamo cosi tornati nello *stato stabile* con S=R=O; Q=1; il BJT saturo e V_{OUT}=0; il condensatore C' alla tensione V_{CC}.
Quando all'istante t_{1+} arriva il fronte di salita del segnale di comando, la situazione è istantaneamente quella riportata in fig 14.
Di conseguenza il diodo resterà interdetto. A regime la situazione diviene quella riportata nella figura 15. Anche in questo caso il diodo è interdetto. La tensione sul condensatore C' tende a zero con costante tempo $\tau''=C' R_1$ che dovrà essere resa piccola, considerando che il tempo che impiega C' a scaricarsi completamente è nell'ordine di $5 \div 6\tau''$. Il resto del circuito rimane nello *stato stabile*.
Nell'istante t_2 il segnale di comando passa dal valore V_{CC} a zero.
Dato che all'istante t_{2-} le tensioni in tutti i nodi del circuito hanno gli stessi valori che avevano all'istante 0, l'andamento delle tensioni tra l'istante t_2 e l'istante $2t_2$ sarà lo stesso di quello già dimostrato tra 0 e t_2.
Questo tipo di multivibratore monostabile *non è retriggerabile* cioè, se durante il tempo in cui l'uscita è alta applichiamo in ingresso un nuovo fronte di discesa, l'uscita resta invariata e non vengono generati nuovi impulsi fino a che il dispositivo non torna nella condizione stabile.
Nelle figure 16, 17 e 18 sono riportati i risultati della simulazione ottenuta con R_1=1KΩ, R_2=1KΩ, C=C'=1μF.

TIMER 555 (CONFIGURAZIONE MONOSTABILE RETRIGGERABILE)

Il circuito di questo temporizzatore è mostrato

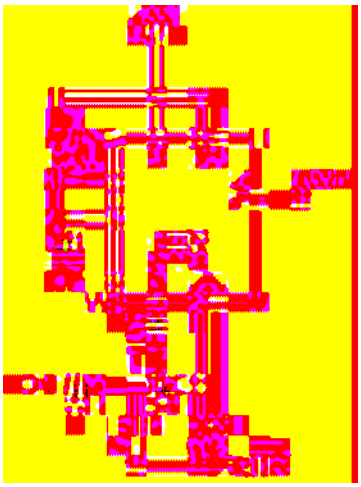


Figura 11 *Schema del timer in configurazione monostabile non retriggerabile*

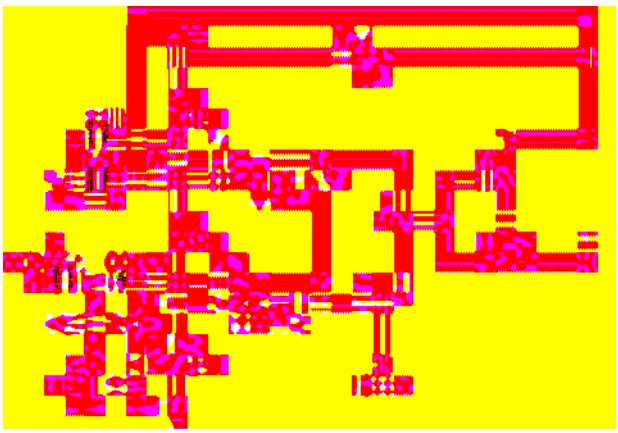


Figura 12 *Schema interno del timer in configurazione monostabile non retriggerabile*

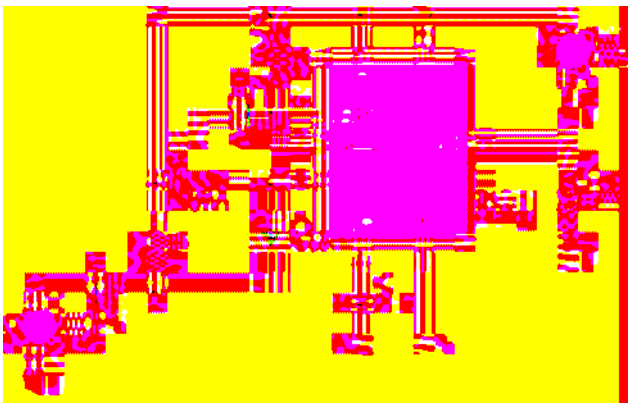


Figura 13 *Schema elettrico del timer in configurazione monostabile non retriggerabile per la simulazione con Cadlogix*

Figura 14 *Rete elettrica collegata al pin 2 quando Vc=Vcc*

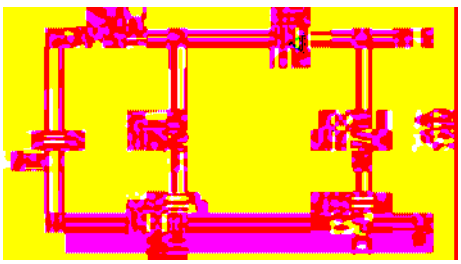


Figura 15 *Rete elettrica collegata al pin 2*

in figura 19, mentre in figura 20 è riportato lo schema elettrico interno.
La dicitura "*retriggerabile*" significa che, una volta innescato l'impulso di uscita, il circuito risente comunque di altri fronti di discesa posti all'ingresso del circuito di comando.
Supponiamo i condensatori inizialmente scarichi e V_{IN} aperto: il diodo si troverà interdetto poiché il suo catodo è a potenziale $+V_{cc}$ mentre l'anodo è a potenziale di massa.
Contemporaneamente il comparatore 2 ha la sua uscita alta (S=1) mentre il comparatore 1 possiede l'uscita a zero (R=O) per cui Q=O e V_{OUT}=1 (= $+V_{cc}$). Il condensatore C si caricherà con costante tempo τ=RC alla tensione $+V_{cc}$; una volta raggiunto il valore di 1/3 V_{cc} il comparatore 2 va a zero (S = 0) per cui l'uscita manterrà lo stato precedente (Q=0, V_{OUT}=$+V_{CC}$) ed il condensatore C continuerà a caricarsi. Quando raggiunge il valore di 2/3 V_{cc} il comparatore 1 porterà la sua uscita alta (R=1), quindi Q=1 quindi V_{OUT}=0. Siamo quindi in una *condizione stabile*: D=Off; V_6=V_2 = $+V_{cc}$; R=1; S=O; Q=1; V_{OUT} = O.
Supponiamo di applicare al circuito di comando una tensione V_{IN} = $+V_{cc}$: istantaneamente il condensatore C' si comporta come un corto circuito ed il dispositivo rimane nella condizione stabile. Nel lungo periodo la situazione rimane invariata poiché la differenza di potenziale ai capi di C' è nulla (su entrambe le armature è presente una tensione pari a $+V_{cc}$).
Se V_{IN} viene portata a zero il condensatore C' si comporta come un cortocircuito per cui il diodo entra in conduzione (dato che il suo anodo (parte p) si trova a $+V_{cc}$ ed il catodo a massa) e il condensatore C, precaricato a $+V_{cc}$, si scaricherà in un tempo brevissimo fino a $+V_{\gamma}$ (con costante tempo τ_1= R_{dir} C). Gli ingressi del flip-flop R ed S si porteranno quindi rispettivamente a 0 e 1 mandando Q a zero e V_{OUT} alta ($+V_{cc}$). Intanto entrambi i condensatori inizieranno a caricarsi e il diodo andrà verso l'interdizione, poiché C' si carica più velocemente di C (in fase di progetto è necessario porre $\tau' << \tau$ dove τ'=R'C). Quando la tensione V_6 = V_c = V_2 raggiunge 1/3 V_{cc} il comparatore 2 commuta a zero (S = 0) e l'uscita V_{OUT} mantiene lo stato

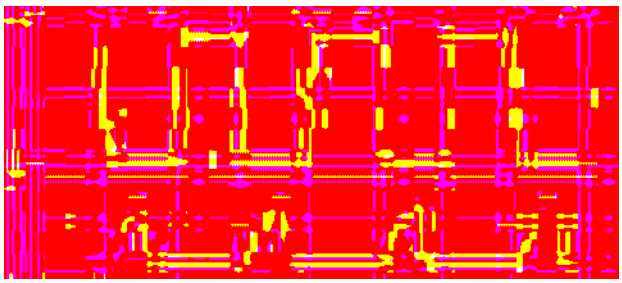


Figura 16 *Forme d'onda ottenute dalla simulazione con CadLogix*

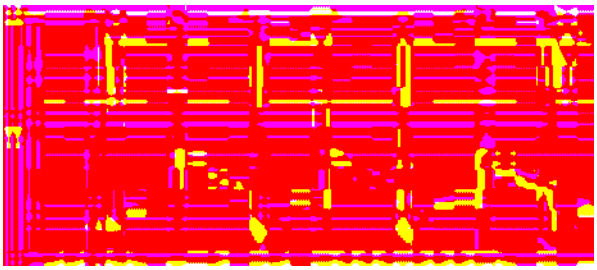


Figura 17 *Forme d'onda ottenute dalla simulazione con CadLogix*

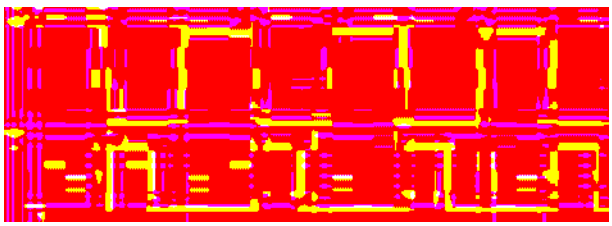


Figura 18 *Forme d'onda ottenute dalla simulazione con CadLogix*

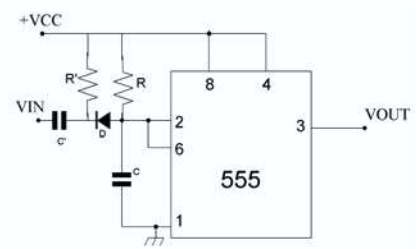


Figura 19 *Schema elettrico del timer monostabile retriggerabile*

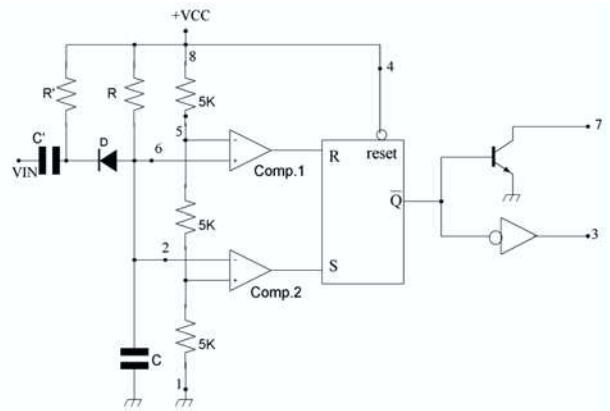


Figura 20 *Schema elettrico interno del timer monostabile retriggerabile*

precedente (R è ancora a zero), mentre una volta raggiunto il valore 2/3 V_{cc}, il comparatore 1 satura positivamente (R=1) e l'uscita V_{OUT} commuta a zero.
La durata dell'impulso sarà data quindi dal tempo che V_6 impiega per andare da V_γ a 2/3 V_{cc}:

$$T_0 = \tau \cdot \ln \frac{3 \cdot (V_{cc} - V_\gamma)}{V_{cc}}$$

Il circuito per la simulazione è quello di figura 21. Nelle figure 22, 23 e 24 sono riportate le forme d'onda ottenute dalla simulazione con CadLogix. Questo tipo di monostabile è *retriggerabile* in quanto, una volta innescato l'impulso d'uscita, il circuito può essere influenzato ad altri eventuali fronti di discesa dell'ingresso del circuito di comando.
Supponiamo di applicare a V_{IN} un fronte di discesa prima che sia terminato l'impulso di uscita progettato. Sul primo fronte di salita dell'impulso di comando la tensione sul catodo del diodo passa istantaneamente a $2V_{cc}$ dato che C' era carico a $+V_{cc}$. Nel lungo periodo tale condensatore tenderà a scaricarsi poiché entrambe le armature sono poste a $+V_{cc}$. Il diodo quindi continuerà a rimanere interdetto e le altre forme d'onda non ne sono influenzate. Sul fronte di discesa dell'impulso di comando invece, il diodo entra in conduzione scaricando il condensatore C alla tensione $+V_\gamma$. Le tensioni $V_{c'}$ e V_6 riprenderanno l'andamento nel modo già illustrato e l'impulso di uscita finirà solo quando la tensione V_6 raggiungerà il valore di 2/3 V_{cc}.
In conclusione possiamo dire che se un impulso di comando cade all'interno della durata dell'impulso di uscita progettato, quest'ultimo avrà una durata pari alla somma di T_0 e il tempo trascorso prima che giungesse il fronte di discesa dell'impulso di comando.

PROSSIMA PUNTATA

Nella prossima puntata tratteremo i circuiti reazionati, con particolare riguardo agli oscillatori a sfasamento. Il simulatore CadLogix, sempre utile in ogni puntata, ci aiuterà nella comprensione del funzionamento di alcuni semplici circuiti.

BIBLIOGRAFIA

1. P. Horowitz, W. Hill – **"The Art of Electronics"**, Cambridge University Press, II ed., 1989.
2. J. Millman, A. Grabel – **"Microelectronics"**, McGraw-Hill International Ed., II Ed., 1987.
3. Datasheet del componente forniti dai diversi Costruttori, disponibili nel sito della Rivista.

Segnalo in particolare quello della Fairchild Semiconductor (www.fairchildsemi.com) che è molto ricco di esempi ed applicazioni.

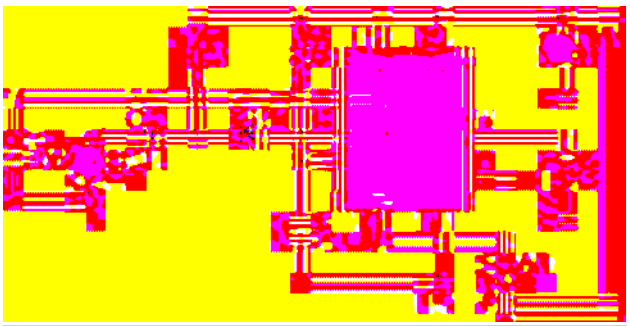


Figura 21 *Schema elettrico del timer monostabile retriggerabile per CadLogix*

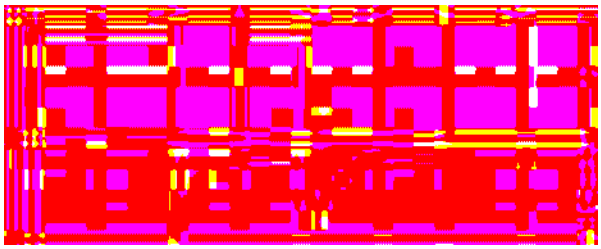


Figura 22 *Forme d'onda ottenute dalla simulazione con CadLogix (monostabile retriggerabile)*

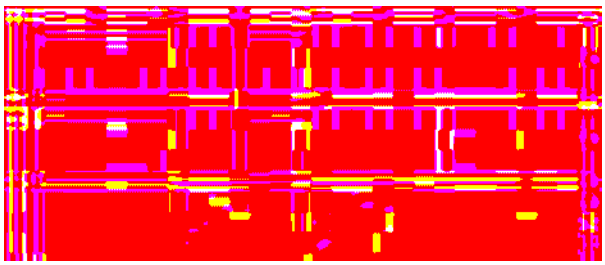


Figura 23 *Forme d'onda ottenute dalla simulazione con CadLogix (monostabile retriggerabile)*

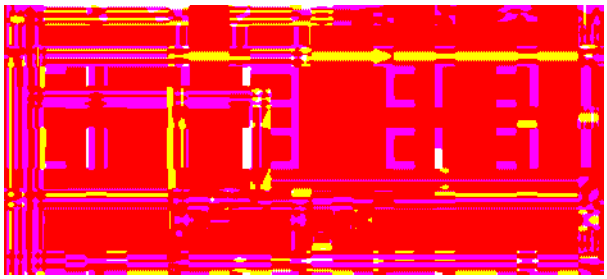


Figura 24 *Forme d'onda ottenute dalla simulazione con CadLogix (monostabile retriggerabile)*

7° Rurale radioamatori, elettronica, home computer

Radio Rivista

26-27 novembre 2005

Padiglione Fiera

La tecnica DDS

Continua la rassegna di progetti utilizzanti la tecnica DDS. In quest'articolo è presentato un riproduttore di suonerie per cellulari, a partire da file in formato RTTTL. Mediante un semplice programma viene generato il file HEX da caricare nel PIC16F84A (massimo 2 suonerie). Si può scegliere la riproduzione singola o continua delle suonerie. Possibili campi d'applicazione: telefonia, avvisatori acustici, elaborazione personale delle suonerie, eccetera.

Ogni giorno la nostra vita è scandita da suoni che provengono dai più disparati oggetti elettronici. L'avvento dei cellulari e l'enorme varietà di suonerie ci ha abituato a pretendere dagli avvisatori acustici suoni sempre più complessi che non siano il semplice beep o la solita sirena bitonale. In quest'articolo è presentato un riproduttore di suonerie, a partire da file in formato RTTL. La tecnica di base è la DDS che ci permette di generare i toni musicali, necessari alla composizione delle suonerie, mediante un micro PIC. Molteplici sono le applicazioni di tale progetto. Per esempio si può pensare di sostituire la stridente suoneria della nostra sveglia con una dolce melodia!

IL FORMATO RTTL

La base di partenza è rappresentata dai files RTTL in cui sono scritte le informazioni relative alla suoneria. In tale file sono riportate il nome della suoneria, la sequenza dei toni musicali e la loro relativa durata. In genere un esempio di un file RTTL è la seguente:

```
Mission:d=4,o=6,b=100:32d,32d#,32d,32d#,32d,
32d#,32d,…
```

La struttura di un file RTTL è:

Titolo suoneria: **d**=durata, **o**=ottava, **b**=battute al minuto: **n1**, **n2**, **n3**...

Dove n1, n2, n3 sono date dall'unione del valore della durata e dal carattere che indica la nota secondo la scala musicale inglese. Nell'esempio sopra riportato con 32*d* s'intende la nota d della sesta ottava (*o=6*) e di durata 32. La corrispondenza tra le note musicali e lettere è data dalla tabella 1.

C	Corrisponde a **DO**
D	Corrisponde a **RE**
E	Corrisponde a **MI**
F	Corrisponde a **FA**
G	Corrisponde a **SOL**
A	Corrisponde a **LA**
B	Corrisponde a **SI**

Tabella 1 *Corrispondenza tra lettere e note musicali*

IL RIPRODUTTORE DI SUONERIE

In questo articolo è presentato un riproduttore di suonerie, implementato mediante un micro PIC16F84A, utilizzando la tecnica di sintesi digitale diretta o DDS. Essenzialmente il riproduttore è un generatore di segnali a durata variabile e a frequenza nell'ambito delle note musicali.

Caratteristiche tecniche del circuito (XTAL =4 MHz)

Tensione d'alimentazione: 2.7-5V

Riproduttore di suonerie

di Salvatore Torrisi
microst@microst.it

Frequenza Toni: Scala musicale standard
Precisione: ±1.7 Hz (XTAL = 8MHz).
Altre caratteristiche elettriche sono quelle riportate sul datasheet del PIC16F84A

Descrizione del progetto

In figura 1 è riportato lo schema elettrico del riproduttore di suonerie a tecnica DDS. Il circuito è costituito da un microcontrollore PIC16F84A e da alcuni componenti passivi. Per la sua realizzazione possono essere usati altri modelli della famiglia PIC16X.

Funzionamento del circuito

Per iniziare/interrompere la riproduzione bisogna agire sul tasto **P1**.
Il circuito permette la riproduzione di due suonerie. Per scegliere la suoneria bisogna agire sul Tasto **P3**.
La riproduzione delle suonerie può avvenire in due modi: singola e continua. Per cambiare modo bisogna premere il tasto **P2**.
In modalità singola la suoneria selezionata è riprodotta una sola volta. Per iniziare la riproduzione bisogna agire su P1 (in tal caso ha la funzione di PLAY). In modalità continua la suoneria selezionata è riprodotta continuamente.
Premendo il tasto **P1** (play) inizia la riproduzione e questa continua fino a che il tasto P1 (stop) è premuto. Ogni volta che il tasto **P2** o **P3** è premuto parte la riproduzione per verificare che il cambiamento voluto.
Nelle tabelle che seguono sono riportate le funzioni dei tasti P1, P2, P3 (tabella 2) e la corrispondente accensione dei led in funzione della funzione selezionata (tabella 3).

Montaggio

In figura 2 e 3 sono riportati il circuito stampato lato rame e lato componenti del riprodutto-

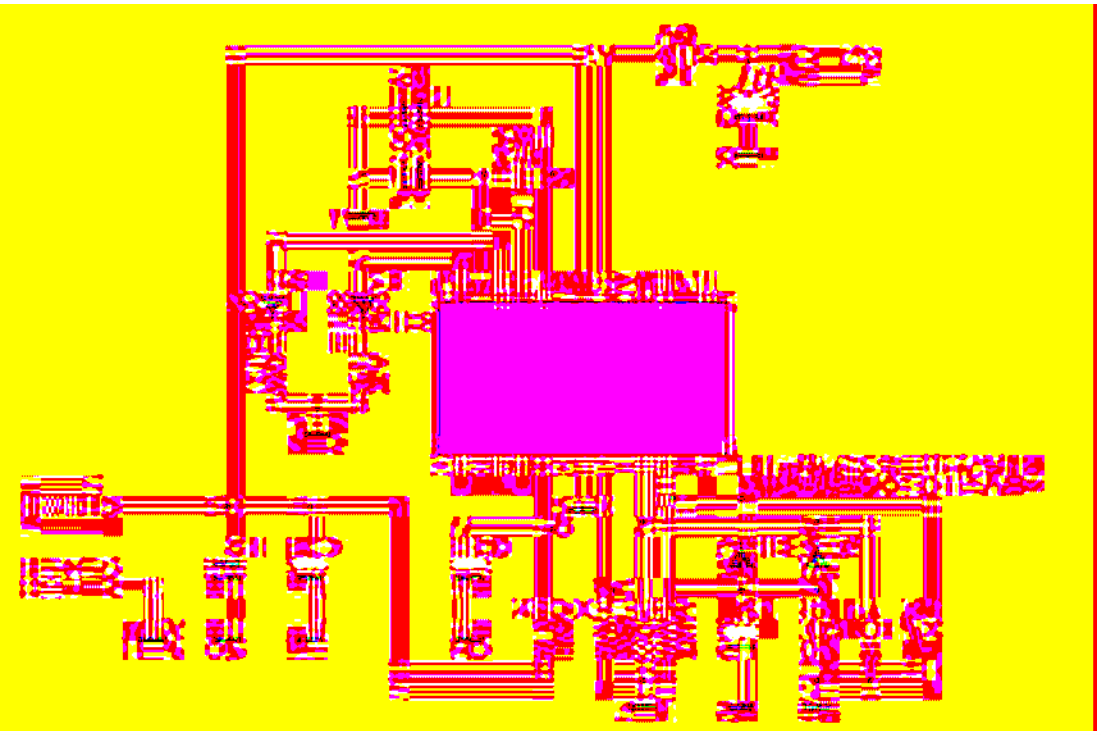


Figura 1 Schema elettrico del riproduttore di suonerie

re. Il circuito puo'essere parte integrante di progetti più grandi e complicati, per questo il circuito stampato è utilizzabile per la realizzazione del solo generatore.

Taratura e Collaudo

La taratura del circuito è semplice ma importante per la qualità della generazione. Per la taratura è necessario utilizzare un frequenzimetro. Dopo aver collegato il probe del frequenzimetro sul pin 15 del PIC, mediante un cacciavite per RF, bisogna agire su C4 ruotandolo in modo da leggere la frequenza del quarzo utilizzato più vicina al suo valore nominale.
Se per qualche motivo non si riesce ad ottenere il valore del quarzo, si può cambiare il valore di C3 o C4. Per ascoltare la riproduzione in uscita si può collegare l'uscita ad un piccolo amplificatore (le casse del PC vanno bene).

Elenco componenti	
Sigla	**Valore**
R1, R2, R3, R4	10 KΩ 1/4 W
R5, R6, R7	1 KΩ 1/4 W
C1, C6	100 nF poliestere
C2	100 µF 16 V elettrolitico
C3	10 pF ceramico
C4	4-20 pF compensatore
C5,C7	1 µF 16 V elettrolitico
D1, D2	IN4148
P1,P2, P3	Pulsanti Normalmente Aperti
IC1	PIC16F84A
XTAL	4 MHz (*)
LD1, LD2	Diodo Led

(*) La scelta del quarzo è molto importante poiché il valore della frequenza **XTAL** rientra nel calcolo d'alcuni parametri delle routine di generazione.

APPLICAZIONI E MIGLIORAMENTI

Lo scopo principale del circuito è di mostrare la possibilità di utilizzare la tecnica DDS per la riproduzione delle suonerie dei cellulari mediante un PIC con un impatto hardware praticamente trascurabile.
Possibili campi d'applicazione sono: telefonia, avvisatori acustici, ripetitori di suoneria, elaborazione personale delle suonerie, eccetera.
Questo circuito può essere la base d'altri progetti più complessi dove implementare tutte quelle funzioni che si vogliono: visualizzazione, impo-

Tasto		Funzione
P1	PLAY/STOP	On/Off riproduzione
P2	MODE	Selezione modalità
P3	SEL	Selezione suoneria

Tabella 2 Corrispondenza tasto-funzione

Funzione	LD1	LD2
PLAY	ON	--
STOP	OFF	--
MODE-SINGLE	--	OFF
MODE-CONT	--	ON
SEL-RING1	--	--
SEL-RING2	--	--

Tabella 3 Corrispondenza funzione - led

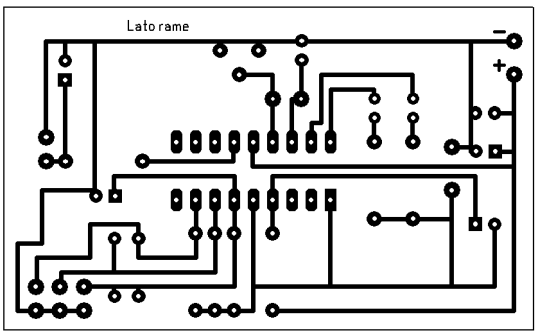


Figura 2 Circuito stampato (lato rame) in scala 1:1

Figura 3 Piano di montaggio

stazione, memorizzazione, eccetera. Un miglioramento potrebbe essere quello di inserire un filtro attivo in uscita al posto di quello passivo (R5-C6) e amplificare l'uscita mediante un IC per amplificazione audio (vedi TBA820 o similari).

LE ROUTINE DEL PROGETTO

Di seguito sarà fatta un'analisi delle principali routine che costituiscono il codice per il PIC.

Ringtone.asm

Il file ringtone.asm è il main program in cui sono definite le variabili, i vari settaggi, gestiti i tasti e la routine di generazione, inclusione dei files suonerie ring1.asm, ring2.asm. In particolare sono implementate le routines per la gestione dei tasti P1, P2 e P3 per l'attivazione, per la modalità di riproduzione e per la selezione della suoneria da riprodurre.

I files ring1.asm, ring2.asm

Questi files contengono la traduzione dei file RTTL in istruzioni per il PIC e sono generati dal programma *RTTL2ASM*. Le informazioni contenute e necessarie per la riproduzione sono: numero (parametro LENGTH), durata e frequenza delle singole note musicali costituenti la suoneria.
Queste informazioni sono restituite mediante l'istruzione RETLW. Ogni singola nota occupa quattro istruzioni RETLW.
Nelle prime tre righe sono memorizzate i valori di H_TIMER, L_TIMER e C_TIMER che definiscono la durata mentre nell'ultima riga è memorizzato il valore da attribuire alla variabile NOTES che definisca la nota secondo la scala musicale.
Nel riquadro seguente è riportato un esempio di un file ringX.asm. Quindi il file conterrà 4x LENGTH righe.

```
; format asm for song canon.txt
; default value:
; name
; beats 80
; octave 5
; duration 4

lengh2   equ 0xe0

song2   org     0x0300
```


Abbonati oggi!

www.farelettronica.com/abbonamento

fare elettronica

CULTURA ELETTRONICA APPLICATA

```
        bsf     pclath, 1
        bsf     pclath, 0
        addwf   pcl, 1
        retlw   0x0 ; valore di h_timer nota1
        retlw   0x0 ; valore di l_timer
        retlw   0x0 ; valore di c_timer
        retlw   0x0 ; valore di notes
        …
        …
        retlw   0x0
```

Il file Tone_DDS.asm

Il file tone_DDS.asm contiene le routines, scritte per il PIC16F84A (ma estendibili ad altri della stessa famiglia), che implementano il generatore di toni musicali usando la tecnica DDS a 16 bit. Queste routines possono essere utilizzate in altri progetti laddove si richiede la generazione di toni musicali. Inoltre questa routine contiene la tabella di corrispondenza tra la FCW (Frequency Control Word) e la frequenza della nota musicale da riprodurre. Per la generazione della singola nota sono necessari due informazioni: durata e frequenza. Queste informazioni sono memorizzate nella file ringX.asm relativo alla suoneria X da riprodurre e devono essere lette e caricate nelle variabili di lavoro della routine. In particolare la durata è caricata nella variabile TIMER a 24 bit realizzata mediante le variabili H_TIMER, L_TIMER e C_TIMER mentre la nota è caricata sulla variabile NOTES. Quindi, quando la routine Tone_DDS è chiamata dal programma principale questa attiva un ciclo di generazione DDS controllato dalla variabile TIMER e dalla variabile FCW1 (H_FCW1, L_FCW1) ricavata dalla tabella di corrispondenza mediante i valore caricato su NOTES. Poiché la suoneria è un insieme di note bisogna eseguire tale operazione per ogni nota della suoneria. La riproduzione quindi avviene nel seguente modo. Si scandisce il file RingX.asm, si caricano le diverse variabili e si attiva il generatore DDS. Tutto questo è ripetuto per il numero di note che sappiamo essere definita dalla variabile LENGTH.

IL SOFTWARE DEL PROGETTO

Per la messa in opera del progetto è fornito sia il software necessario per la generazione del file .hex da caricare nel PIC, sia una collezione di suonerie scritte in formato RTTL (vedi il file *RTTTL.zip*). Il programma principale è il *RTTL2ASM.exe*.
Questo programma genera i file musicali in formato assembler convertendo le informazioni scritte nel file RTTL in una serie d'istruzioni eseguibili dal Micro.

Generazione del file .hex

I passi necessari per la generazione del file *ringtone.hex* con cui programmare il PIC sono:

1. Creare una directory di lavoro.
2. Scompattare in essa il file *ringtone.zip* fornito con il progetto.
3. Scompattare il file *RTTTL.zip* (una raccolta di suonerie in formato RTTTL) nella stessa directory di lavoro.
4. Lanciare il file ringtone.bat.
5. Partirà il programma *RTTL2ASM.exe*.
6. Immettere il numero N di suonerie che si vuole riprodurre (massimo 2 per il PIC16F84A).
7. Scegliere dalla lista (premere L per visualizzarla) il file (.txt) da riprodurre.
8. Saranno generati file ring1.asm, ring2.asm, e così via.
9. Sarà lanciato il programma (mpasm.exe) che compila il file ringtone.asm generando il file ringtone.hex, se non ci sono errori.
10. Caricare *ringtone.hex* nel PIC mediante i soliti programmi usati per la programmazione dei PIC come ICPROG, eccetera.

WEB LINK

Tutto il materiale necessario alla realizzazione del progetto è contenuto nel file scaricabile dal sito di Fare Elettronica.

CONCLUSIONI

Come ulteriore implementazione della tecnica DDS, presentato nel numero 240 (Giugno 2005) di Fare Elettronica, abbiamo presentato un riproduttore di suonerie a partite dai file RTTL.
Questo progetto evidenzia la potenzialità di tale tecnica che permette di ottenere ottimi risultati con un basso consumo di risorse hardware e software.
Presto presenteremo altri progetti utilizzanti questa tecnica interessante.

25° MARC

mostramercato attrezzature
radioamatoriali & componentistica
hardware • software
ricezione satellitare
editoria specializzata
radio d'epoca

Fiera di Genova
17-18 Dicembre 2005

sabato ore 9 • 18,30
domenica ore 9 • 18

ENTE PATROCINATORE:
A.R.I. - Ass. Radioamatori Italiani
Sezione di Genova
Salita Carbonara 65 b - 16125 Genova
C. P. 1117 - 16121 Genova - Tel./Fax 010.25.51.58
www.arigenova.it

ENTE ORGANIZZATORE E SEGRETERIA:
STUDIO FULCRO s.a.s.
Piazzale Kennedy, 1 - 16129 Genova
Tel. 010.561111 - Fax 010.590889
www.studio-fulcro.it e-mail: info@studio-fulcro.it

Introduzione alle

Le memorie Flash sono oggi impiegate in una grande varietà di dispositivi elettronici: si va dai dispositivi più diffusi come telefoni cellulari, fotocamere digitali e lettori MP3, fino ad arrivare ai più complessi sistemi di controllo dell'automobile e sistemi di diagnostica medica. In questo articolo introdurremo i concetti fondamentali delle memorie Flash: partendo dal funzionamento del singolo dispositivo e dalle operazioni di memoria si passerà all'affidabilità, alle architetture e alle applicazioni.

Le memorie sono componenti fondamentali nell'elettronica perché in esse vengono conservati i dati, e nel caso dei circuiti di elaborazione, anche le istruzioni. Tra i vari tipi di memoria, negli ultimi anni si sono sviluppate le memorie Flash perché possiedono numerosi vantaggi: dimensioni ridotte, affidabilità, elevate capacità di immagazzinamento. Questi vantaggi hanno fatto convergere gli sforzi dei produttori di semiconduttori verso questo tipo di memoria, dapprima con intensi programmi di ricerca e sviluppo, e in seguito con la produzione e la commercializzazione delle stesse e dei dispositivi elettronici che ne fanno uso.

LE MEMORIE FLASH

Le memorie Flash sono dispositivi a stato solido, programmabili e cancellabili elettricamente, che conservano l'informazione anche se viene interrotta l'alimentazione. Queste memorie uniscono i vantaggi delle RAM (Random Access Memory, memoria ad accesso casuale) e delle ROM (Read Only Memory, memoria di sola lettura), eliminandone contemporaneamente le limitazioni. Infatti le RAM sono programmabili e cancellabili elettricamente ma perdono l'informazione dopo l'interruzione dell'alimentazione, e le ROM non perdono l'informazione senza alimentazione ma possono essere programmate solo una volta. Per queste caratteristiche, le memorie Flash rientrano in una più vasta categoria, le memorie non volatili, di cui fanno parte le EEPROM (Electrically Erasable Programmable Read Only Memory, memoria di sola lettura cancellabile e programmabile elettricamente). In effetti, le memorie Flash sono l'evoluzione delle EEPROM; rispetto a queste ultime richiedono meno area (quindi garantiscono una maggiore integrazione e in definitiva un costo minore), e hanno un procedimento di programmazione più semplice e veloce.

DESCRIZIONE DELLA CELLA FLASH

La cella Flash ha una struttura simile a quella di un transistor MOS (Metal Oxide Semiconductor), pertanto cominceremo dalla descrizione di quest'ultimo. Il transistor MOS (figura 1) è costituito da un substrato di silicio con drogaggio di tipo p che è la base su cui è costruito l'intero dispositivo, il gate è in polisili-

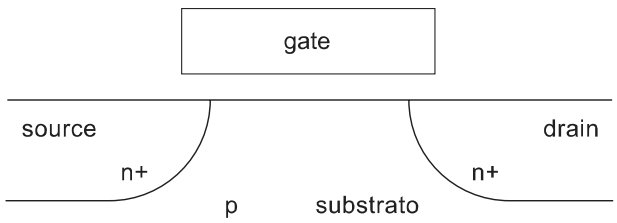


Figura 1 Il transistor MOS

memorie Flash

di Teodoro Fristachi
(teodoro.fristachi@tin.it)

cio e le diffusioni di drain e source sono di tipo n^+, dove il segno "+" indica una alta concentrazione di drogaggio (vedi speciale "Come nasce un circuito integrato - Fare Eletronica 243 pag. 40). Il transistor appena descritto è a canale n, perché la conduzione avviene sfruttando gli elettroni che scorrono dal source al drain. La conduzione di corrente nel canale avviene in questo modo: fissata la tensione V_{DS} ad un valore per esempio pari a 2 Volt, la corrente I_d assume valori significativamente diversi da zero quando la tensione di gate V_G è maggiore della tensione di soglia V_t (figura 2). La tensione di soglia di un MOS (e anche di una cella Flash) è pertanto un parametro fondamentale che caratterizza il dispositivo.
Esiste anche il MOS di tipo p, che si ottiene invertendo i drogaggi del MOS di tipo n, ma in questo articolo ci occuperemo solo di quest'ultimo tipo.

Passiamo adesso alle descrizione della cella Flash. Come detto, la cella di memoria Flash è molto simile a quella di un MOSFET ma in più è presente un secondo gate che è flottante (floating gate), cioè non contattabile elettricamente (figura 3).

Il floating gate è separato dal control gate dal dielettrico di interpoly che è un triplo strato di ossido di silicio – nitruro di silicio – ossido di silicio (ONO) caratterizzato da un'alta costante dielettrica per migliorare l'accoppiamento capacitivo tra control gate e floating gate. L'altro ossido, quello che separa il floating gate dal substrato determina le proprietà della cella Flash, ed è chiamato ossido di tunnel. È un ossido di silicio cresciuto termicamente di altissima qualità, ed ha un ruolo fondamentale nelle operazioni di memoria (lettura, programmazione, cancellazione). La struttura appena descritta dà origine a numerosi accoppiamenti capacitivi, come si può vedere nel modello circuitale di figura 4.

Il floating gate è l'elemento caratterizzante della cella Flash perché in esso possono essere immagazzinati elettroni. Questa è la chiave che permette di realizzare una memoria a stato solido: a seconda se il floating gate è carico di elettroni oppure no, la cella di memoria si trova nello stato "0" o nello stato "1". Poiché il floating gate non è contattabile elettricamente, gli elettroni restano al suo interno oppure al di fuori di esso anche se viene interrotta l'alimentazione, al contrario delle memorie RAM che perdono l'informazione se si interrompe l'alimentazione. Come si fa a vedere se nel floating gate è pre-

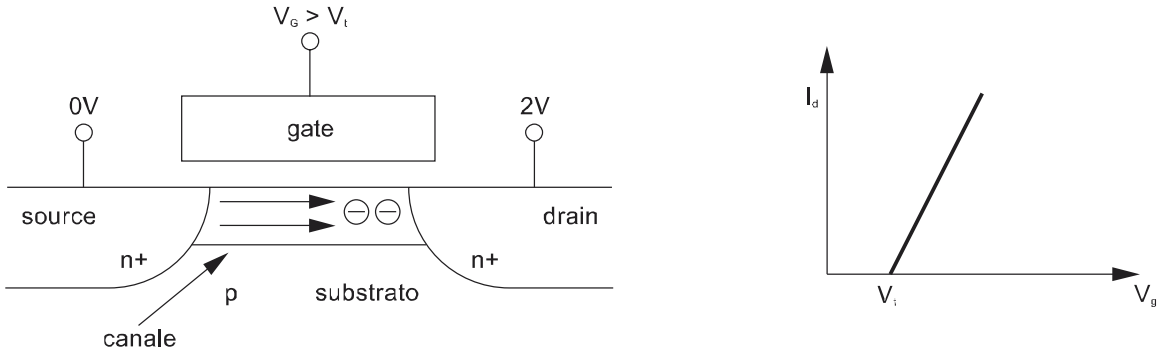


Figura 2 *La conduzione nel transistor MOS*

sente carica oppure no? Basta misurare la classica caratteristica I_d-V_{GS}. Questa curva ha lo stesso andamento di quella del transistor MOS, ma a seconda se è presente carica o no sul floating gate la curva risulterà traslata orizzontalmente (figura 5). Nelle applicazioni reali esiste una circuiteria apposita che rileva quale delle due curve caratterizza la cella, e da questa misura si riesce a sapere se la cella di memoria si trova nello stato "0" o nello stato "1".

Facendo sempre riferimento alla figura 5 si vede che lo spostamento della caratteristica I_d-V_{GS} equivale allo spostamento della tensione di soglia V_t della cella. Lo spostamento della tensione di soglia è dato dalla relazione:

$$\Delta V_t = \frac{|Q|}{C_{CF}}$$

essendo |Q| la carica immagazzinata e C_{CF} la capacità tra control gate e floating gate.

Se nel floating gate non è presente carica, la cella è nello stato "cancellato" (stato logico "1"); se invece vi è carica (negativa), la cella è nello stato "programmato" (stato logico "0"). Le corrispondenti tensioni di soglia prendono il nome di "tensione di soglia nello stato cancellato" e "tensione di soglia nello stato programmato". Nel ricavare queste curve, la tensione V_{DS} viene mantenuta ad un valore fisso convenzionale (pari ad esempio a 2V).

Mentre la tensione di control gate è quella che si applica dall'esterno, la vera tensione che cade sull'ossido di tunnel e che quindi determina le operazioni di lettura, scrittura e cancellazione della memoria è la tensione di floating gate, che risulta essere inferiore a quella del control gate secondo la relazione:

$$V_{FG} = \alpha_G \bullet V_{CG}$$

dove α_G è un fattore di accoppamento capacivo

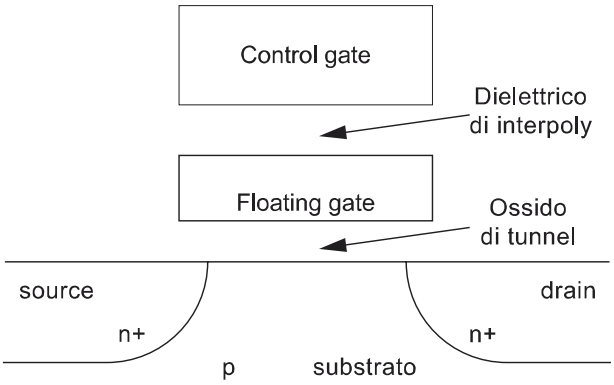


Figura 3 La cella Flash

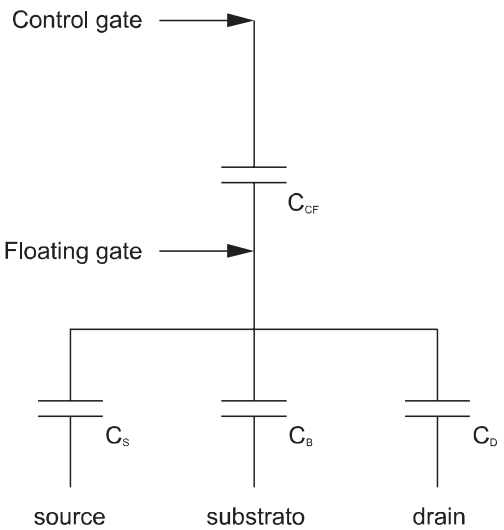


Figura 4 Circuito equivalente della cella Flash

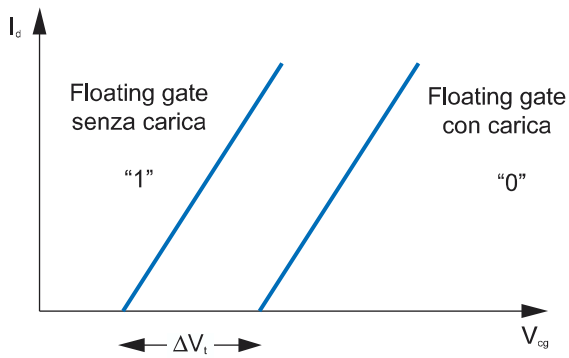


Figura 5 Spostamento della tensione di soglia nella cella Flash

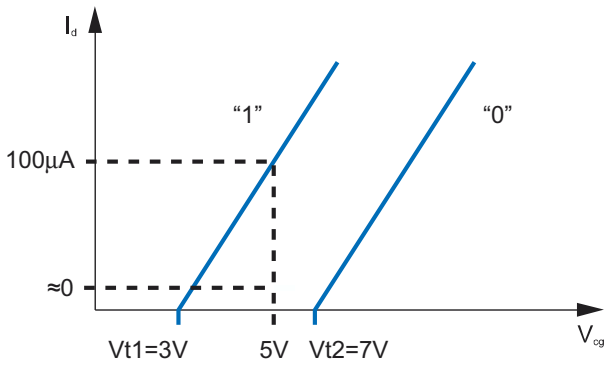


Figura 6 Lettura della cella Flash

che vale circa 0.6 ed è dato dalla relazione:

$$\alpha_G = \frac{C_{CF}}{C_T}$$

essendo C_{CF} la capacità tra control gate e floating gate e C_T è la capacità totale (è la somma delle quattro capacità di figura 4).

OPERAZIONI DI MEMORIA: LETTURA, PROGRAMMAZIONE, CANCELLAZIONE

Lettura

Questa operazione deve rivelare se la cella è nello stato programmato o cancellato. Per esempio, nella tecnologia 0.18 µm (dove, appunto, la minima risoluzione è di 0,18 milionesimi di metro), la tensione V_{DS} si fissa ad 1V, la V_{CG} si pone in un valore intermedio tra le soglie degli stati "1" e "0": se lo stato "1" si trova sui 3V e lo stato "0" sui 7V, la tensione di lettura si pone a 5V (figura 6). In corrispondenza di queste tensioni scorre una corrente il cui valore identifica lo stato della cella. Valori tipici di queste correnti sono ≈100 µA nello stato cancellato e ≈0 nello stato programmato.

Programmazione

Questa operazione inietta carica nel floating

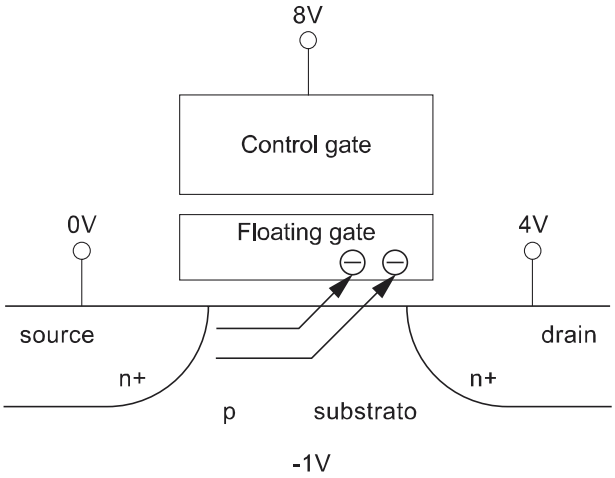


Figura 7 Programmazione della cella Flash

gate e porta la cella nello stato programmato "0". Le tensioni tipiche di una programmazione sono riportate in figura 7, per una tecnologia 0.18 µm. Il metodo tradizionale di programmazione è per CHE (Channel Hot Electrons), che consiste nell'accelerare gli elettroni longitudinalmente nel canale tramite una tensione V_{DS} tra drain e source, e poi di iniettarli nel floating gate tramite la tensione trasversale V_{CG}. Gli elettroni ricevono energia dalla tensione V_{DS} (si "riscaldano", per questo prendono il nome di hot electrons), ed in seguito è sufficiente una tensione di qualche Volt tra canale e floating gate affinché vadano ad immagazzinarsi nel floating gate. Naturalmente la tensione al control gate è tale da tenere conto anche del coefficiente α_G di accoppiamento.

Cancellazione

È l'operazione complementare della programmazione: il suo compito è riportare gli elettroni dal floating gate al substrato, facendo così ritornare la cella di memoria nello stato cancellato "1".
La cancellazione avviene per tunnel di Fowler-Nordheim. L'effetto tunnel è descrivibile tramite la meccanica quantistica e consiste nel passaggio di elettroni attraverso una barriera di potenziale (in questo caso la barriera è l'ossido di tunnel). L'espressione della densità di corrente che scorre dal control gate al substrato durante il tunnel è la seguente:

$$J_{fn} = A \cdot E_{ox}^2 \cdot \exp\left(\frac{-B}{E_{OX}}\right)$$

dove A e B sono costanti e E_{OX} è il campo elettrico sull'ossido da tunnel. Le unità di misura sono Ampere/cm². Per ottenere l'effetto tunnel si deve applicare un elevato campo elettrico sull'ossido di tunnel (che deve il suo nome proprio a questo effetto) tramite una elevata tensione V_{CG} , mentre la V_{DS} vale 0V (figura 8).
Questa elevata tensione permette agli elettroni immagazzinati nel floating gate di superare lo spessore dell'ossido di tunnel e di tornare nel substrato. Da notare che con le tensioni applicate come in figura 8 il tunnel avviene lungo tutto il canale e non solo alla giunzione di source, come avveniva in passato (source erase).

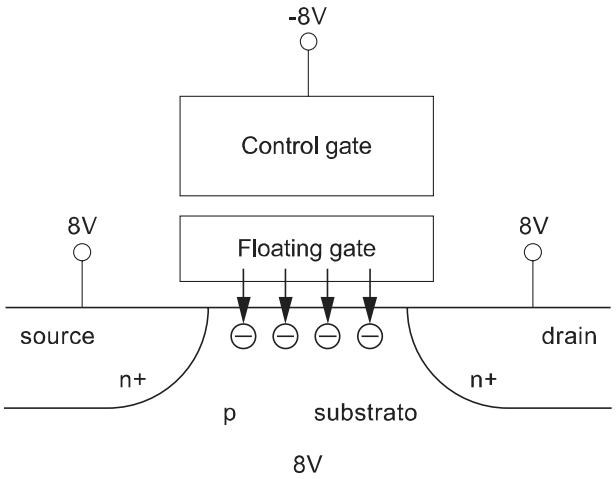


Figura 8 Cancellazione della cella Flash

ANCORA SULLA TENSIONE DI SOGLIA

Finora abbiamo parlato di tensioni di soglia nello stato cancellato e nello stato programmato lasciando intendere che queste tensioni siano

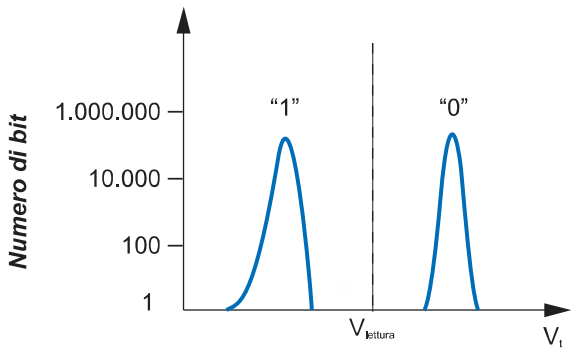


Figura 9 Distribuzioni delle tensioni di soglia

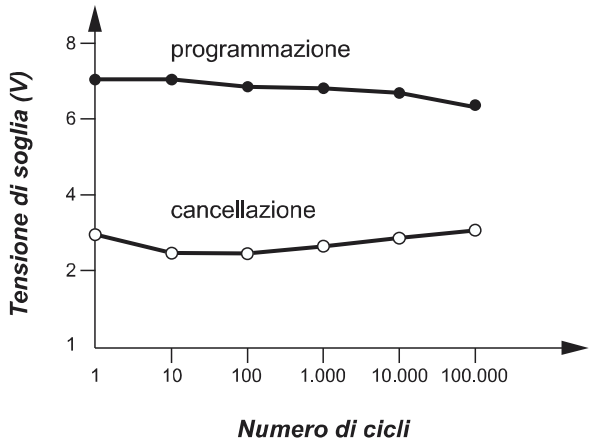


Figura 10 Chiusura della finestra delle tensioni di soglia

sempre le stesse. In realtà la tensione di soglia che si raggiunge a causa della carica che va nel floating gate dopo una programmazione è leggermente diversa dalla tensione di soglia della programmazione precedente. La stessa cosa avviene per la cancellazione. In effetti è più corretto parlare di distribuzioni delle tensioni di soglia, come si vede in figura 9. Le tensioni di soglia nello stato programmato e cancellato presentano uno spread intorno ad un valore medio, ma comunque rimangono confinate nelle distribuzioni come in figura.
Si può notare che la distribuzione delle celle cancellate non è simmetrica e presenta una coda esponenziale sulla sinistra. La coda rappresenta una larga popolazione di celle che si cancellano più velocemente del bit tipico. Questa popolazione è troppo grande per essere attribuita solo ai difetti: si pensa che sia correlata anche a variazioni statistiche della carica nell'ossido e alla struttura degli elettrodi.

AFFIDABILITÀ DELLE MEMORIE FLASH

L'affidabilità per le memorie Flash si fonda su due direttive:

- Capacità di funzionamento dopo almeno 100.000 cicli di programmazione/cancellazione (endurance).
- Ritenzione del dato per almeno 10 anni (data retention).

ENDURANCE

Le specifiche di affidabilità standard impongono che la cella sia in grado di funzionare per almeno 100.000 cicli di programmazione/cancellazione. L'ossido di tunnel è l'elemento critico per l'affidabilità delle memorie Flash a causa del suo ridotto spessore, e la programmazione e la cancellazione a lungo andare degradano questo ossido perché producono difetti microscopici nella sua struttura molecolare: ciò si può riscontrare nella chiusura della finestra delle tensioni di soglia (figura 10). La chiusura della finestra delle tensioni di soglia significa che le tensioni di soglia dello stato programmato e cancellato si avvicinano, e il rischio è che i circuiti di lettura della memoria non riescano a riconoscere correttamente il bit. Nella curva della V_t nello stato cancellato si nota un iniziale abbassamento: questo è dovuto all'intrappolamento di cariche positive che facilitano il tunnel per la cancellazione, a lungo termine invece prevale l'intrappolamento di elettroni e la presenza di difetti nell'ossido carichi negativamente e la curva si alza.
La ciclatura influisce negativamente sulle prestazioni in generale, a cominciare dai tempi di programmazione e cancellazione (figura 11).

DATA RETENTION

La cella di memoria, secondo le specifiche, deve conservare la carica nel floating gate per almeno 10 anni dopo la programmazione. La diminuzione di solo 10% della carica immagazzinata può portare ad errori di lettura. Le cause che portano alla perdita di carica sono due: i difetti nel dispositivo e le operazioni stesse di pro-

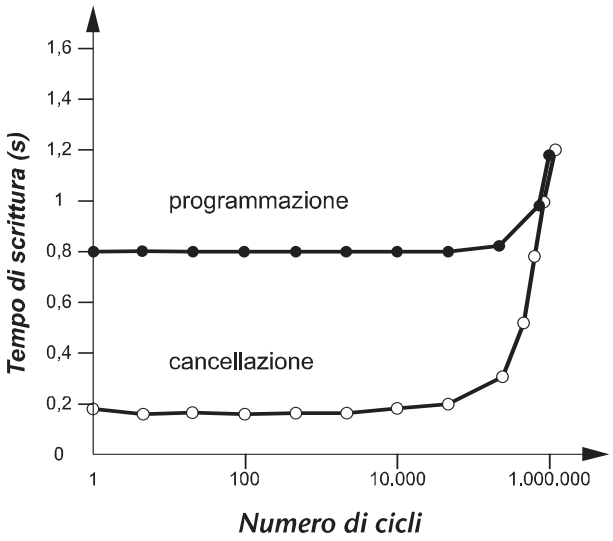


Figura 11 Peggioramento delle prestazioni di una cella a seguito del prolungato funzionamento

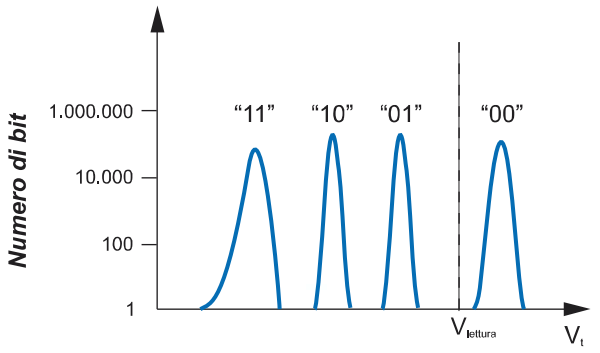


Figura 12 Distribuzioni delle tensioni di soglia nel caso Multilevel

grammazione e cancellazione. Per avere un ordine di grandezza, il floating gate non può perdere più di una decina di elettroni al giorno affinché il dato non sia compromesso. Per valutare la ritenzione si usano test accelerati con alti campi ed alte temperature: per esempio portando la memoria per 500 ore ad una temperatura di 250 °C, si ottiene la stessa degradazione che si otterrebbe portando la memoria per 10 anni a 80 °C. Si attribuisce infatti alla corrente di perdita di gate attraverso l'ossido successiva a ciclatura la responsabilità principale della perdita di elettroni dal floating gate.

CENNI SULLE MEMORIE FLASH MULTILEVEL

Il concetto di memoria Flash Multilevel è nato per aumentare il numero di bit immagazzinabili in una singola cella e, in effetti, è il metodo più efficiente per scalare le dimensioni effettive del silicio occupato da 1 bit. In una cella a singolo bit sono state definite due tensioni di soglia: quella dello stato cancellato e quella dello stato programmato. Nel caso Multilevel per immagazzinare N bit servono 2^N tensioni di soglia. Per esempio per 2 bit servono 4 livelli di soglia (figura 12).
Per avere quattro stati, la finestra totale delle tensioni di soglia è più larga, ma la finestra tra due stati vicini diventa più stretta: è necessario avere distribuzioni più strette per le soglie. Come conseguenza il circuito di lettura deve essere più raffinato, dato che i margini di rumore diminuiscono, e la maggiore finestra impone delle tensioni più alte, con aumenti dei disturbi di lettura e peggioramento dell'affidabilità. È facile intuire come i problemi implementativi aumentino in modo considerevole, ma l'aumento della densità di immagazzinamento è tale da giustificare questa crescente complessità.

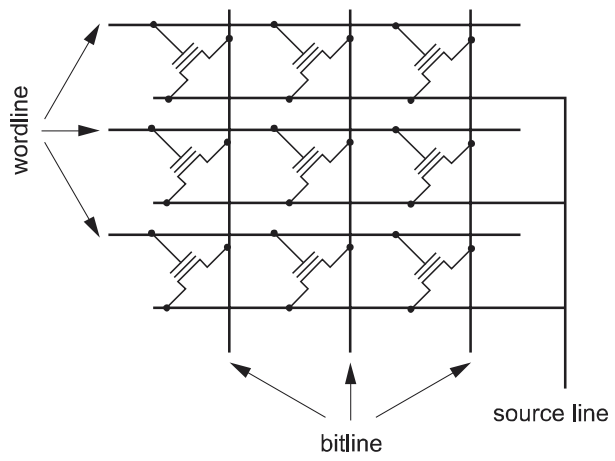


Figura 13 Architettura NOR

ARCHITETTURE DI MEMORIE FLASH

Finora ci siamo occupati del funzionamento della singola cella Flash, ma una memoria completa è composta da numerose celle Flash e dai circuiti necessari alla gestione della memoria stessa. Per poterne illustrare il funzionamento è necessario collocarsi ad un livello di astrazione più elevato, pertanto da adesso analizzeremo le memorie dal punto di vista dell'architettura.
Le architetture più diffuse sono la NOR e la NAND, qui di seguito descritte.

ARCHITETTURA NOR

L'architettura NOR è usata in una vasta gamma di applicazioni che richiedono buone prestazioni e media densità d'integrazione. La sua organizzazione è a matrice, con righe (wordline) e colonne (bitline) come riporato in figura 13.
Tutti i gate delle celle in una riga sono connessi alla stessa wordline mentre tutti i drain delle celle in una colonna sono connessi alla stessa bitline; i source di tutte le celle in un settore sono connessi ad una linea di source comune. Questa organizzazione è ripetuta 8 o 16 volte, ottenendo in questo modo un byte o una word.

ARCHITETTURA NAND

Nell'architettura NAND (figura 14) il source di una cella è connesso drain al della cella successiva: i tempi d'accesso (sensing) risultano più lenti perché l'accesso all'informazione è seriale; inoltre le tensioni di programmazione e cancellazione sono molto alte (possono arrivare anche a 20V): la presenza di queste tensioni pone seri vincoli di progetto e può provocare problemi di affidabilità del dispositivo. Il vantaggio di questa architettura è la dimensione molto piccola della cella. Per esempio, in un processo 0.4 µm, l'area è 1.47 μm^2 per una cella NOR e solo 1.1 μm^2 per una cella NAND, con il risultato di una maggiore densità di immagazzinamento. Questo tipo di architettura viene usata principalmente per applicazioni di immagazzinamen-

SISTEMI DI VIDEOSORVEGLIANZA

WIRELESS

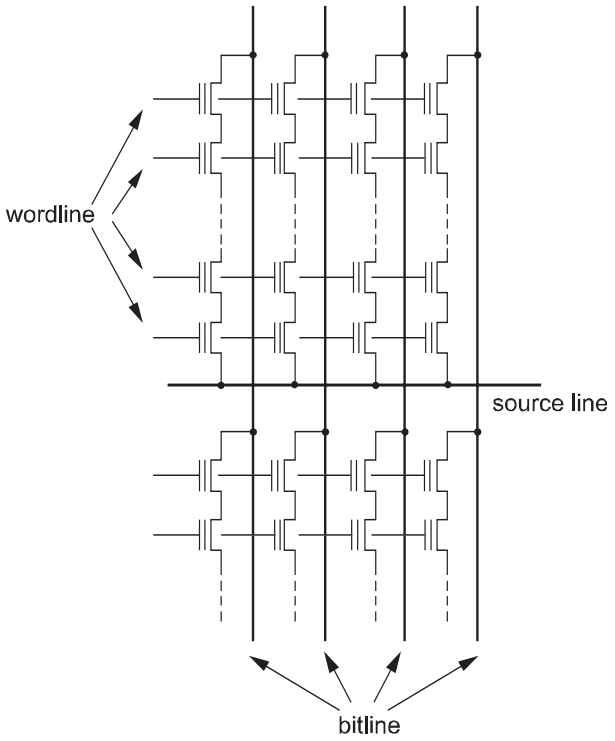


Figura 14 Architettura NAND

to di massa, in cui il tempo di accesso alla singola cella non è di fondamentale importanza.

LETTURA

La lettura viene realizzata in due passi fondamentali: decoding e sensing. Il decoding indirizza la singola cella di memoria all'interno della matrice e deve gestire le differenti tensioni per leggere, programmare e cancellare; per ottenere ciò c'è una prima fase di "predecoding" che seleziona la cella di memoria e poi c'è la fase di decoding che fornisce le tensioni necessarie al gate e al drain delle celle (figura 15). I decoder sono dei multiplexer a transistor nMOS (MOS a canale n) i cui ingressi provengono dai predecoder.

Il sensing è probabilmente l'operazione più critica in una memoria Flash, e il suo scopo è leggere l'informazione immagazzinata nella cella, cioè distinguere se la cella è programmata o cancellata. Lo schema fondamentale di sensing considera le correnti della cella da leggere e di una cella di riferimento e le converte in tensioni, poi le confronta tramite un comparatore decidendo lo stato della cella. La figura 16 è uno schema semplificato di una architettura differenziale di sensing. I due rami sono speculari: su uno c'è la cella Rif di riferimento sull'altro la cella Sel selezionata nella matrice di memoria. La cella Sel è polarizzata con le tensioni V_{GS} e V_{DS} dai decoder di riga e di colonna, poi la sua corrente viene convertita dal blocco Conv in una tensione che va in ingresso al comparatore. L'altro ingresso del comparatore proviene dalla cella Rif di riferimento che è polarizzata dalla stessa V_{GS} della cella sotto misura e da una V_{DS} fissa fornita dal blocco Pol.

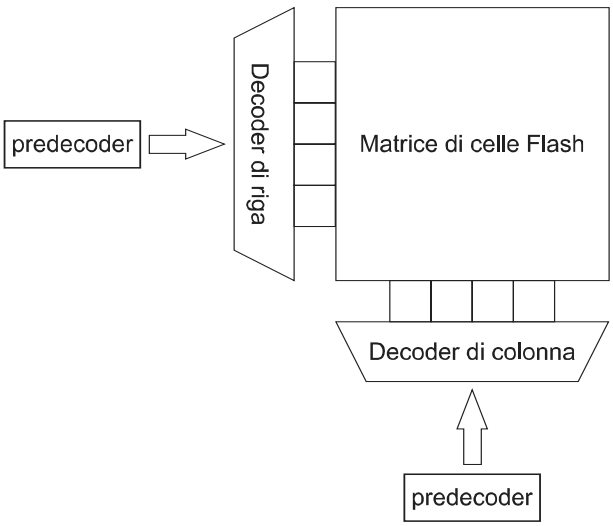


Figura 15 Indirizzamento della memoria

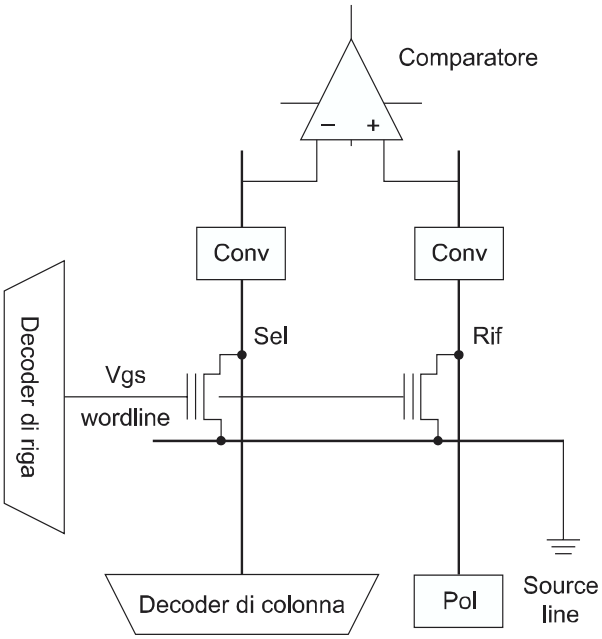


Figura 16 Architettura differenziale di sensing

ORGANIZZAZIONE IN SETTORI

Quando una cella è indirizzata per la programmazione, tutti i gate della wordline sono polarizzati: si parla in questo caso di disturbo di programmazione dovuto allo stress del gate. Alla lunga questa situazione potrebbe dare problemi per l'affidabilità, per questo motivo le memorie vengono suddivise in settori di dimensione fissa, che va da 64 KB fino ad arrivare ai 512 KB. Le dimensioni standard delle memorie attualmente in produzione sono 64 e 128 MB e per raggiungere capacità maggiori vengono affiancati più blocchi assieme.
Un altro motivo per la divisione in settori è che la cancellazione viene eseguita su un intero settore e non sul singolo byte: nelle memorie Flash infatti non sono presenti i transistor di selezione per ogni cella come nelle EEPROM, in questo modo la cancellazione non può essere effettuata sul singolo byte ma in cambio ci sono dei vantaggi più importanti: l'area del chip è minore e la velocità di cancellazione è maggiore.

PROBLEMI DI BASSA TENSIONE DI ALIMENTAZIONE

Le tensioni di alimentazione tendono a diminuire seguendo lo scaling dei dispositivi con lo scopo di abbassare il consumo di potenza. Valori standard per le tensioni di alimentazione attualmente sono 3.6 V e 2.7 V. Poiché le tensioni per le diverse operazioni di memoria sono in genere più alte di queste tensioni di alimentazione, si devono realizzare circuiti appositi per generare tali valori. Se la memoria è di tipo Double Supply, il suo package presenta due pin dedicati alle alimentazioni V_{PP} e V_{CC}, e la più alta delle due (V_{PP}) è utilizzata per eseguire tutte le operazioni ad alte tensioni. Se invece la memoria è Single Supply (un solo pin dedicato all'alimentazione), è necessario generare le tensioni più alte tramite un circuito di boosting (per la lettura) o tramite pompe di carica (per la programmazione e la cancellazione).

PROGRAMMAZIONE

Come già discusso nel paragrafo precedente, se la memoria è Double Supply le alte tensioni provengono dal pin V_{PP}, se è Single Supply provengono da pompe di carica. A causa di queste alte tensioni sono necessari circuiti di regolazione per evitare transizioni spurie che possano danneggiare la cella o ridurre la sua affidabilità dopo numerosi cicli di programmazione/cancellazione. Inoltre sono necessarie altre tensioni per le operazioni di verifica dopo la programmazione. Lo schema a blocchi del program path è in figura 17. Nei latch di ingresso ci sono i dati e gli indirizzi delle celle da modificare; nel blocco Confronto dati si effettuano le verifiche prima e dopo la programmazione: se non è andata a buon fine è possibile ripetere il ciclo di programmazione al massimo per un numero prefissato di volte, oltre il quale l'algoritmo si arresta con un errore. Il buffer di uscita porta le tensioni direttamente sui pin del package. Le tensioni alle righe giungono attraverso il regolatore V_{RR}, e alle colonne le tensioni giungono tramite il regolatore V_{RC}. Da notare che le tensioni di gate variano a seconda dell'operazione che si sta effettuando: lettura, programmazione, cancellazione o verifica (tabella Tensioni di gate). Per gestire le varie tensioni, nel program path viene posizionato uno switch subito dopo il regolatore della tensione di gate V_{RR} (figura 18).

Tensioni di gate	
Operazione	**V_{GATE}**
Lettura	V_{CC}
Programmazione	V_{PP} (10÷12V)
Verifica programmazione	V_{PV} (≈7V)
Cancellazione	< V_{CC}
Verifica cancellazione	V_{EV} (≈3V)

CANCELLAZIONE

L'algoritmo di cancellazione prevede dapprima la programmazione di tutte le celle del settore, per portare le soglie sullo stesso valore, e successivamente la sua cancellazione, che può essere ripetuta più volte se l'algoritmo di verifica rileva che non tutte le celle sono state subito cancellate.
Nel caso di memorie Double supply la source line del settore da cancellare deve essere portata da 0 Volt fino a Vpp e deve mantenere questa tensione per tutta la durata dell'impulso, mentre deve rimanere a 0 Volt durante la lettura e la programmazione. Il circuito che garanti-

sce questo funzionamento è il source switch, che è collegato direttamente alla source line (figura 19).
Nel caso di memorie Single Supply le alte tensioni per la cancellazione sono generate da pompe di carica (figura 20).

Il parallelo di condensatori costituisce una pompa di carica positiva che funziona in due fasi. Durante la fase 1 (parte superiore della figura) i condensatori C1 e C3 sono a tensione Vcc, e questi stessi condensatori trasferiscono carica positiva ai condensatori C2 e C4, rispettivamente attraverso gli interruttori i2 e i4. Durante la fase 2 (parte inferiore) Vcc trasferisce carica positiva al condensatore C1, il condensatore C2 a C3, e C4 all'uscita, attraverso gli interruttori i1, i3 e i5. La carica così accumulata innalza la tensione e ad ogni periodo l'incremento è pari a Vcc.
Lo stesso principio vale per le pompe di carica negativa. La tensione in uscita è direttamente proporzionale al numero degli stadi e alla capacità dei condensatori.
Un valore tipico per la V_{PP} è 12V, e questo valore massimo è imposto dalla struttura stessa della pompa di carica: si dovrebbe avere un numero elevato di condensatori per avere più stadi e quindi una tensione più elevata, ma ciò significherebbe un notevole dispendio di area.
Un altro svantaggio delle pompe di carica è che possono erogare basse correnti (sarebbero più alte con capacità più grandi, ma anche in questo caso si ritorna al problema dell'area), che inoltre sono ancora più basse se le tensioni sono alte.
Anche in questo caso per evitare transizioni spurie dovute alle alte tensioni le memorie Single Supply adottano un regolatore per la tensione di uscita dalla pompa di carica; è inoltre presente un limitatore che disabilita la pompa se la tensione diviene troppo alta.
Infine, un'ultima osservazione sulla tecnologia costruttiva: per poter utilizzare al control gate una tensione negativa come nel caso della cancellazione (figura 8), è necessaria una tecnologia triple-well (substrato di tipo p – n-well – p-well – n-well).
Infatti se non ci fosse la triple-well a garantire l'isolamento, la giunzione drain-substrato del transistor nMOS presente nell'inverter che si connette alla wordline sarebbe polarizzata direttamente.

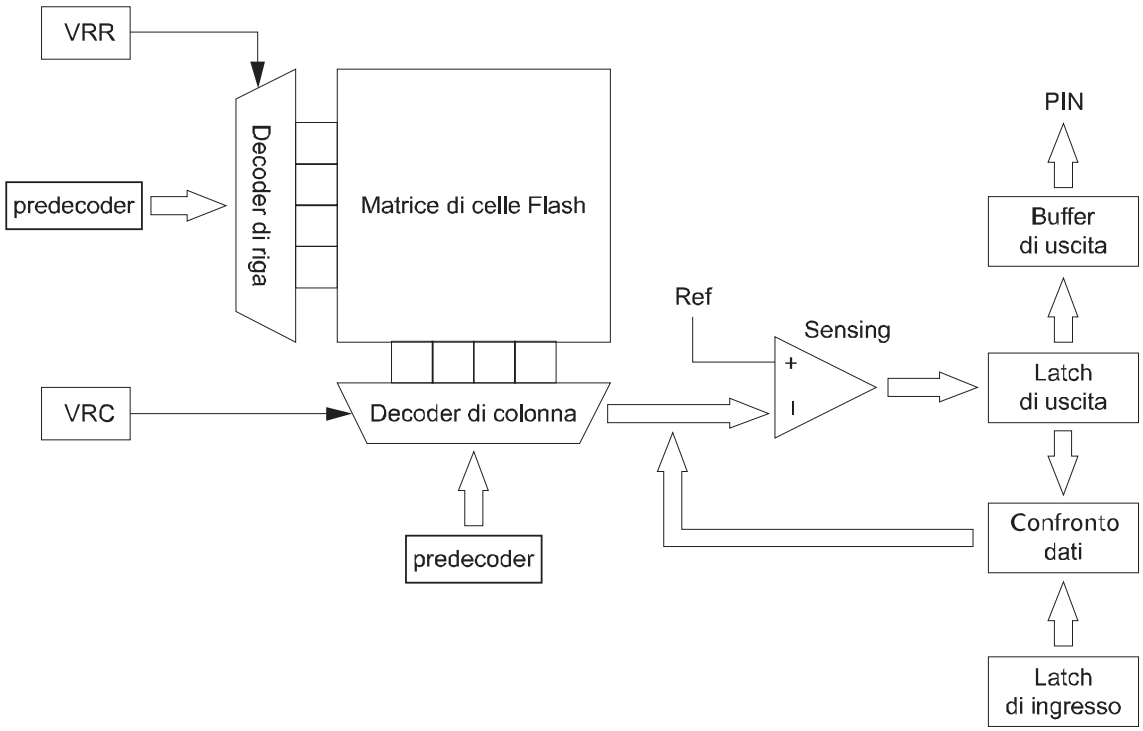


Figura 17 Program path

LOGICA DI CONTROLLO

La logica di controllo per le operazioni più lente (programmazione e cancellazione) è inclusa direttamente nel chip della memoria Flash (embedded logic).
In questo modo l'unità di controllo non è appesantita dalla gestione di compiti che richiedono molto tempo. Inoltre l'interfaccia risulta semplificata: tutti gli algoritmi di programmazione, cancellazione, verifica, timing sono trasparenti all'utente, che deve solo fornire l'operation code e può facilmente controllare lo stato della memoria tramite i pin esterni.

DISPOSITIVI IN COMMERCIO E APPLICAZIONI

Le memorie Flash sono presenti in gran parte dei dispositivi elettronici che usiamo tutti i giorni, in pratica dove c'è un apparecchio elettronico si è quasi sicuri di trovare una memoria Flash. Buona parte delle applicazioni che prenderemo in esame di fatto sono diventate realizzabili solo dopo lo sviluppo delle memorie Flash.
In generale, l'architettura NAND è utilizzata per le applicazioni che richiedono grandi quantità di memoria come ad esempio pen drive, lettori MP3, Flash card, mentre l'architettura NOR è usata dove si deve fare elaborazione dei dati, come nei sistemi di gestione e controllo.
I pen drive sono ideali per trasportare dati grazie alla interfaccia USB e alle dimensioni ridotte. Spesso includono un lettore MP3, e arrivano fino alla capacità di 1GB.
Le flash card sono schede di memoria che vengono utilizzate nelle fotocamere digitali. Esistono vari formati: Secure Digital, CompactFlash, MultiMediaCard, xD-Picture Card, Memory Stick e SmartMedia. I vari tipi di card sono incompatibili, e il tipo da utilizzare dipende dalla fotocamera digitale. Le capacità arrivano fino a 4GB, e presto si arriverà a 8 GB.
Esistono anche mini hard disk realizzati con memorie Flash: hanno dimensioni massime inferiori ai cinque centimetri e capacità fino a 6 GB, non hanno parti meccaniche né magnetiche come gli hard disk tradizionali e sono utilizzati in alcuni modelli di videocamera digitale.
Nel resto del paragrafo descriveremo alcune applicazioni delle memorie Flash. Nei dispositivi citati qui di seguito le memorie interagiscono con molti altri componenti elettronici, tuttavia gli innumerevoli schemi possibili si possono ricondurre allo schema generale di figura 21.

Nelle automobili e più in generale nei veicoli da strada le memorie Flash trovano applicazione nei sistemi per la sicurezza, come ad esempio airbag e ABS, e nel sistema di gestione del motore nonché nel sistema di navigazione satellitare.
Per quanto riguarda i computer, le memorie Flash si trovano nei BIOS, nelle stampanti, nei modem ADSL e nei router, solo per citare le applicazioni più note.
Nel campo dei dispositivi portatili, le memorie Flash sono indispensabili per i telefoni cellulari, in particolare per quelli di terza generazione che possono trasmettere e ricevere file multimediali, e sono inoltre presenti anche nei computer palmari (PDA).
Le memorie Flash si trovano anche nei ricevitori

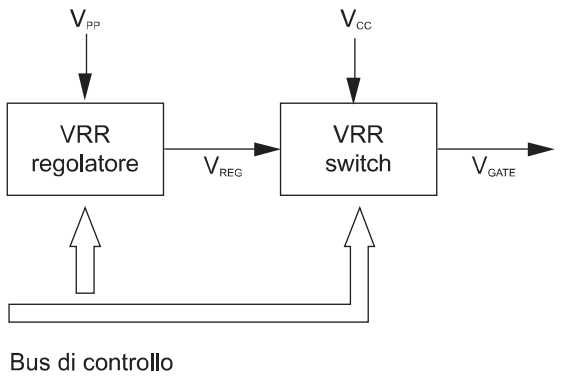


Figura 18 Regolatore e switch per le tensioni di gate

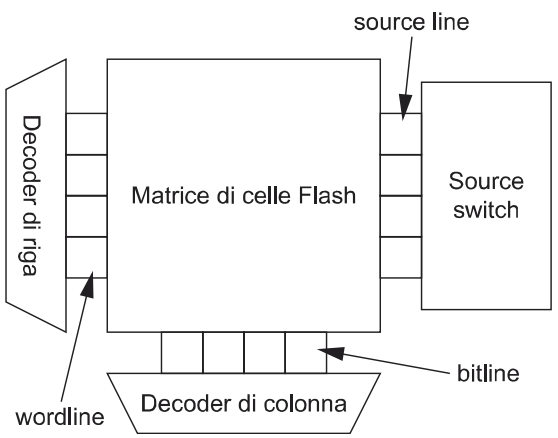


Figura 19 Switch per le tensioni di source

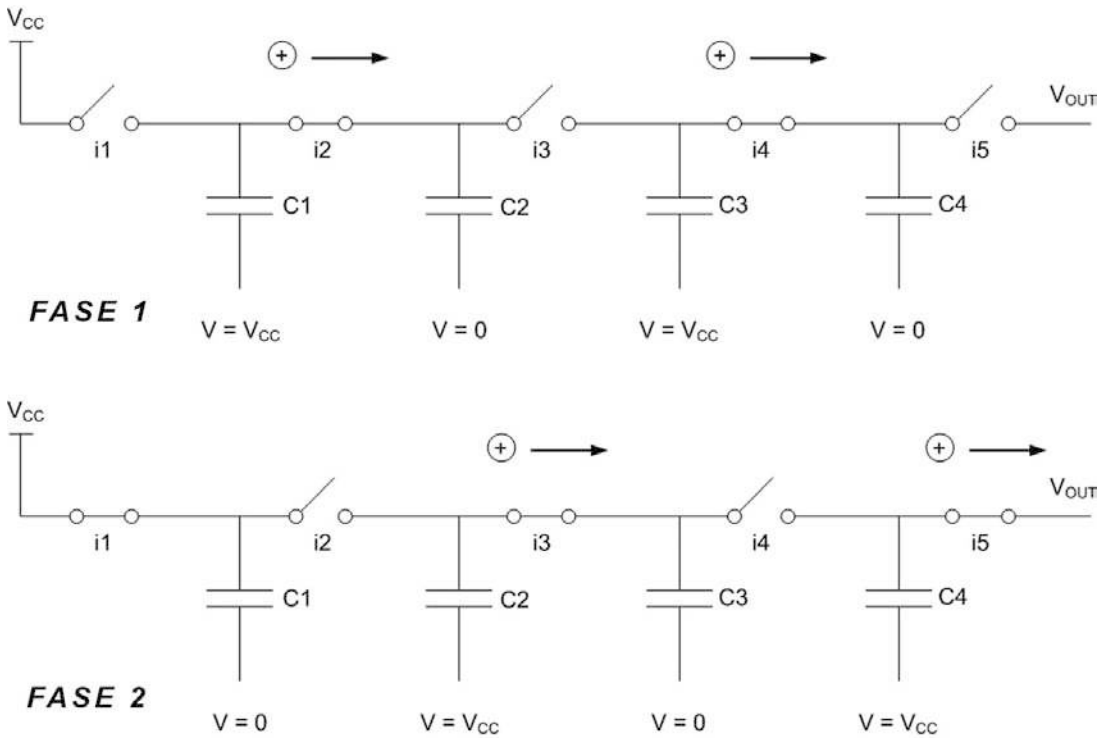


Figura 20 Pompa di carica positiva

satellitari e nei decoder, nei lettori di codice a barre e nei terminali di vendita.
In medicina, sono presenti nei defibrillatori portatili, pacemaker, misuratori di pressione, analizzatori, sistemi che producono immagini tramite risonanza magnetica e sistemi per il monitoraggio delle funzioni vitali dei pazienti.

CONCLUSIONI

In questo articolo abbiamo esaminato le memorie Flash da vari punti di vista: per cominciare abbiamo studiato la struttura della cella Flash, le operazioni di memoria e l'affidabilità, poi abbiamo continuato con le architetture e gli schemi a blocchi dei circuiti esterni alla memoria e infine abbiamo passato in rassegna alcune delle numerose applicazioni.
Naturalmente sono molti gli aspetti che non abbiamo trattato, ed alcuni sono stati solo sfiorati come ad esempio il tunnel Fowler-Nordheim, ma ciò è inevitabile dato il carattere introduttivo di questo articolo il cui scopo è illustrare il funzionamento delle memorie Flash e spiegare i motivi per cui sono così diffuse. Il mondo delle memorie Flash è in continua evoluzione e la ricerca è molto attiva per trovare nuove e più efficienti soluzioni: ad esempio la densità di integrazione crescerà e l'impiego delle memorie Multilevel sarà sempre più diffuso.
Le memorie Flash rappresentano una parte importante del mercato dei semiconduttori, ed in futuro troveranno sempre più vasta applicazione nei settori in cui sono attualmente utilizzate e troveranno nuove applicazioni in settori sempre più diversificati.

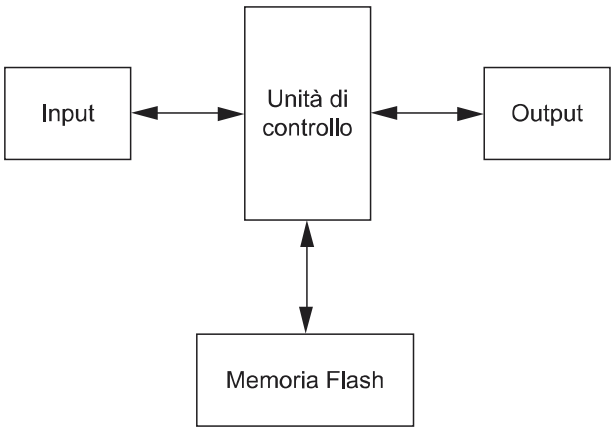


Figura 21 Architettura generale di un dispositivo elettronico con memoria Flash

Ritorna a Ud

la fiera dell'elettronica

& del radioamatore

Con oltre 150 espositori

e tanta informatica ai prezzi

più bassi d'Italia

MOSTRA MERCATO

10-11

DICEMBRE 2005

prevendita biglietti dalle ore 08.30

- COMPUTER
- ELETTRONICA
- RADIANTISMO
- EDITORIA
- TELEFONIA
- TV-SATELLITARE
- HOBBISTICA

Organizzato

COMPRADIO

Sesta parte
n° 244 - Ottobre 2005
Musika maestro! Generazione di melodie, note, effetti sonori e musica con il PIC

Settima parte
n° 245 - Novembre 2005
Pilotiamo i motori passo-passo

Ottava parte
n° 246 - Dicembre 2005
Ambiente di sviluppo

Mikrobasic per PICmicro

In questa puntata affrontiamo la programmazione e la gestione dei motori passo-passo utilizzando il Mikrobasic. Una scelta obbligata per intraprendere lo studio dell'automazione e della robotica in particolare.

L'utilizzo dei motori passo-passo, detti anche motori stepper, può sembrare difficoltoso. In effetti senza l'utilizzo di un microcontrollore diventa davvero difficile utilizzarli, soprattutto per quanto riguarda la scelta dei componenti elettronici da adottare. Grazie al PIC, certi compiti decisionali divengono subito facilmente risolvibili: basta infatti progettare un'adeguata scheda di pilotaggio ed il software farà tutto il resto.

BREVE INTRODUZIONE AI MOTORI PASSO-PASSO

A differenza dei normali motori a corrente continua o alternata, i motori passo-passo sono utilizzati nelle applicazioni dove è richiesta la massima precisione di movimento e di spostamento angolare. Esempi tipici, peraltro molto diffusi, sono le stampanti, i disk driver, i bracci meccanici, i foto-copiatori, gli scanner, eccetera.
Come dice il nome stesso, si tratta di motori che, con opportuni impulsi pilotati, ruotano il proprio asse di un determinato angolo. Il vantaggio principale consiste nella possibilità di programmare non solo la velocità di rotazione ma anche il numero di giri richiesti, il momento e la posizione esatta di partenza e di arrivo. Una programmazione quindi che consente la creazione di dispositivi con un'altissima precisione nei movimenti.
Se prendete in mano un motore passo-passo, e ruotate lentamente il suo albero, avvertirete subito un andamento a scatti (e non lineare e fluido come i normali motori). Ebbene, sono proprio questi i passi (step) che il motore rispetta nel suo avanzamento.
Nell'acquisto di un motore passo-passo, un primo dato caratteristico è appunto il numero di step necessario per completare un giro completo dell'asse; esistono infatti motori che compiono pochi step (quindi poco precisi) e motori che ne compiono moltissimi per completare una rivoluzione intera (quindi molto precisi). Nella tabella sottostante riportiamo i gradi di rotazione di un singolo passo (**S.A.**) ed il corrispondente numero di passi necessari ad effettuare un giro completo (**S.P.R.**).

S.P.R.	S.A.
10	36°
12	30°
20	18°
24	15°
40	9°
48	7,5°
50	7,2°
100	3,6°
200	1,8°
400	0,9°

Come si può notare, i due dati sono inversamente proporzionali: più elevato è il numero di step necessari a formare un giro completo e più "piccolo" è l'arco che deve compiere il motore. In altre parole, più l'angolo dello step è piccolo, più è preciso il motore, in quanto aumenta la "risoluzione" del passo.
Nella tabella successiva elenchiamo, in linea di massima, i vantaggi e gli svantaggi che derivano utilizzando simile motore, nelle applicazioni pratiche.

Pilotiamo i motori passo-passo

di Giovanni Di Maria
(g.dimaria@farelettronica.com)

A fronte di tanti piccoli svantaggi, è sicuramente consigliabile utilizzare un motore passo-passo per l'elevatissima precisione di rotazione, che i normali motori non potrebbero mai fornire, se non utilizzando sofisticati encoder, freni digitali ed ottici che, sicuramente, ne farebbero lievitare a dismisura il prezzo.

Vantaggi
Riproducibilità di qualsiasi movimento
Possibilità di cambio di direzione senza modifica dell'hardware
Massima precisione di movimento
Coppia elevata
Possibilità di recupero da periferiche guaste
Possibilità di bloccare il motore in una posizione precisa
Velocità impostabile a piacere
Possibilità di impostare velocità bassissime
Minima usura di utilizzo
Svantaggi
Alto costo relativo
Difficoltà di reperimento in commercio
Difficoltà maggiore nel suo pilotaggio
Esigenza di una scheda supplementare per il pilotaggio ad alta potenza
Impossibilità ad ottenere velocità elevatissime
Rotazione non fluida ma con piccolissimi intervalli

Schema base di un motore passo-passo

Un motore è formato internamente da tanti induttori che devono essere alimentati sequenzialmente per assicurare il corretto movimento dell'asse. Qualunque sia il numero di induttori interni, esso è sempre rappresentato, per comodità, da quattro bobine poiché, alla fine, sono altrettante il numero delle fasi.

Come si pilota un motore

Senza approfondire la teoria dei motori passo-passo per i nostri esperimenti ci serviremo di quelli *unipolari*, certamente più semplici da comandare. La tecnica principale consiste nel pilotare un avvolgimento per volta, sequenzialmente, in modo che il motore avanzi di uno *step* per volta. Come vedrete, appena avrete tra le mani un motore, il compito più arduo e difficile sarà proprio quello del riconoscimento dei fili di alimentazione, solitamente tutti di colore diverso.

Riferiamoci alla figura 2, che riporta la tipica struttura di un motore passo-passo. Per prima cosa occorre collegare il filo (o i fili) contrassegnati da Vcc al positivo dell'alimentazione. Per fare avanzare il motore di un passo per volta

Figura 1 *Tipico motore passo-passo*

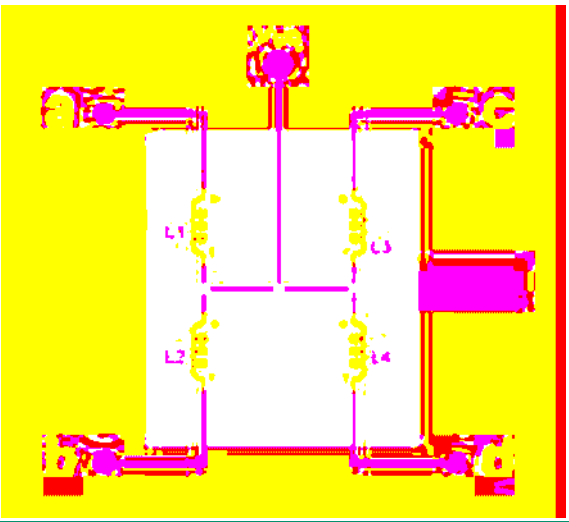


Figura 2 *Struttura di principio*

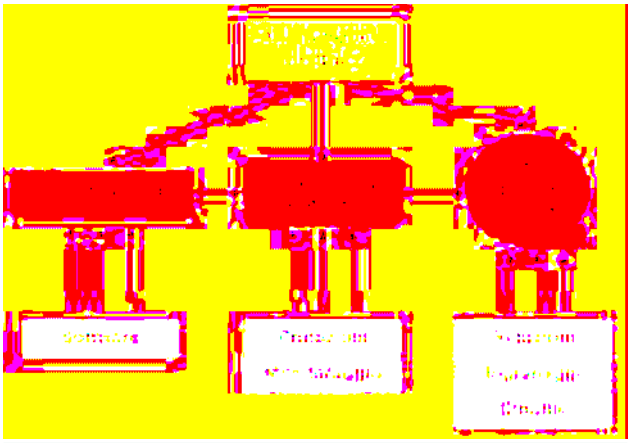


Figura 3 *Schema a blocchi del sistema base*

occorre, a questo punto, collegare alternativamente **a massa** i quattro terminali, contrassegnati con le lettere A-B-C-D. Solo se riuscite a trovare la "corretta" sequenza di pilotaggio si potrà far "girare" un motore stepper.

REALIZZIAMO LA SCHEDA DI PILOTAGGIO

Un motore passo-passo è un dispositivo *di potenza*, per cui non può essere direttamente pilotato dal microcontrollore.
Occorre realizzare un'interfaccia che trasformi i deboli impulsi del PIC in segnali a corrente elevata utilizzabili dal motore stesso.

Perché l'interfaccia

L'insieme di tutti i dispositivi che realizzeremo costituisce un valido esempio di sistema esperto adibito al controllo e pilotaggio di periferiche di potenza e getta sicuramente le basi a tutte quel-

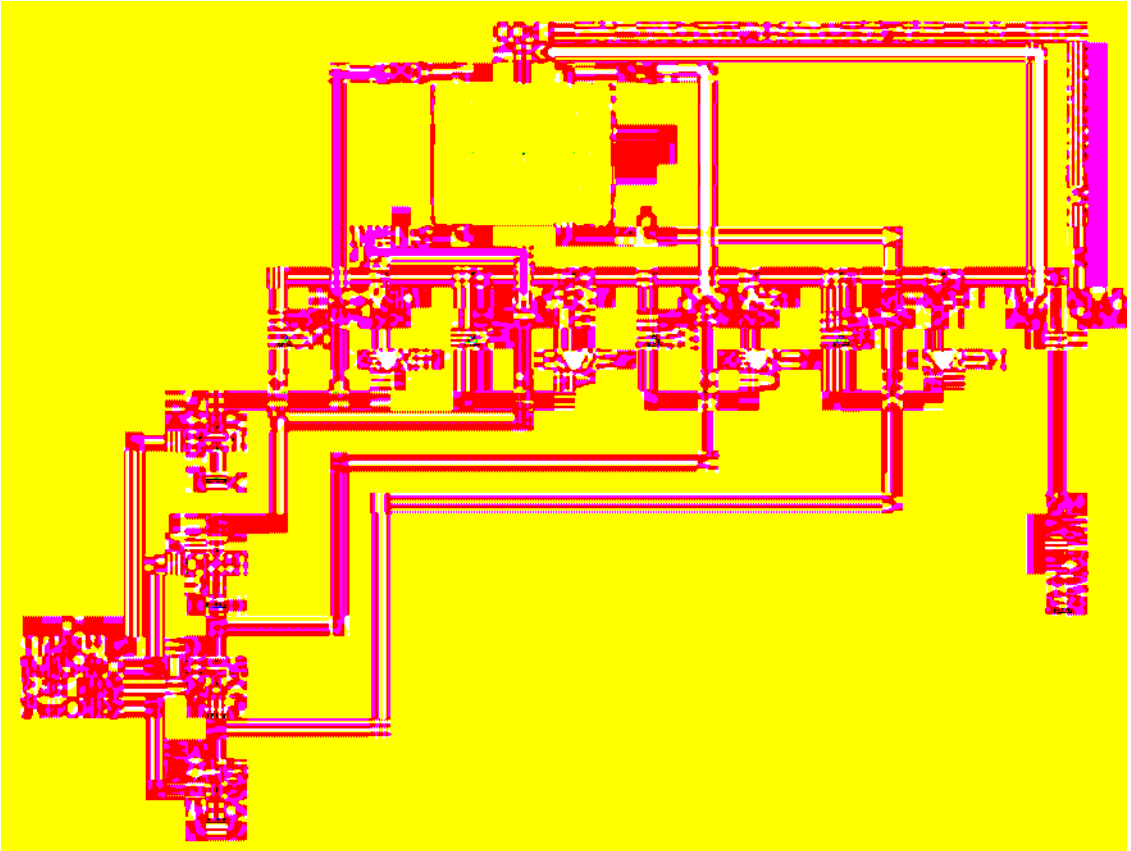


Figura 4 *Schema elettrico dell'interfaccia*

le attività in cui è la macchina, e non l'uomo, a svolgere i compiti più disparati, come avviene, ad esempio, nella robotica.
Il sistema base da concepire è formato da tre unità logiche fondamentali, come evidenziato nello schema a blocchi ci cui in figura 3:

- Il microcontrollore, che rappresenta il cervello e che dirige le azioni per il pilotaggio del motore.
- L'interfaccia di potenza, che trasforma i deboli impulsi del microcontrollore in impulsi più "robusti" per il pilotaggio del carico.
- Il motore vero e proprio.

Schema elettrico

Analizziamo adesso lo schema elettrico dell'interfaccia di potenza, illustrata in figura 4.

Il circuito viene alimentato dal generatore G1, una batteria o un alimentatore robusto di 5 o più Volt (sino ad un massimo di 30 Volt, dipendente dalla tensione del motore passo-passo).
Gli impulsi a bassa potenza, provenienti dal microcontrollore, sono applicati ai quattro ingressi rappresentati dai morsetti X4 (1 e 2) e X5 (1 e 2). Tali impulsi sequenziali vanno direttamente a pilotare i mos-power IRF 540, dei veri e propri di cavalli di battaglia (dati di targa: 100 Volt D-S, 0.077 ohm Rgs, 28 A Id). Tali Mosfet si comportano pertanto come veri e propri interruttori di potenza, senza inerzia e contatto meccanico, che abilitano o disabilitano il passaggio di corrente attraverso le quattro bobine del motore. Dal momento che ogni induttore, quando è attraversato da corrente pulsante, genera delle extra-tensioni, sono stati inseriti quattro diodi di protezione atti ad annullare i picchi pericolosi e rappresentati dai diodi D1, D2, D3 e D4.
Infine, gli impulsi ad onda quadra provenienti direttamente dal PIC sono visualizzati dai quattro diodi led, limitati in corrente dalle rispettive resistenze R1, R2, R3 e R4.
Come si nota dallo schema, il motore passo-passo si collega in questo modo: il filo comune si connette al positivo dell'alimentazione (tramite il morsetto X3) ed eventualmente (tratteggiato) anche il secondo filo comune (nel caso di motore a 6 fili) mentre i fili delle quattro bobine

si collegano ai quattro morsetti (X1-1, X1-2, X2-1, X2-2). I colori dei fili del motore dello schema sono puramente indicativi.
È importante sottolineare un aspetto fondamentale dei mos-power: il gate lavora esclusivamente in tensione (a differenza della base dei normali transistor BJT che lavorano in corrente) ed è ad altissima impedenza. Questo vuol dire che se toccate il terminale di ingresso con un dito, potreste attivare la chiusura del canale Drain Source.

Il circuito stampato ed il montaggio

Occorre adesso approntare il circuito stampato, utilizzando possibilmente la tecnica della fotoincisione, con la quale si ottengono migliori risultati. Realizzato il prototipo occorre eseguire la foratura con punta di trapano da 0,8 mm oppure da 1 mm. Si può dunque procedere alla saldatura dei componenti elettronici, iniziando naturalmente da quelli di dimensioni più ridotte, quali resistenze e diodi, per poi passare a quelli di profilo maggiore, come i mos-power, i diodi led ed i morsetti.
In figura 6 è riportata la traccia in dimensioni naturali del circuito stampato, mentre in figura 7 la disposizione dei componenti.

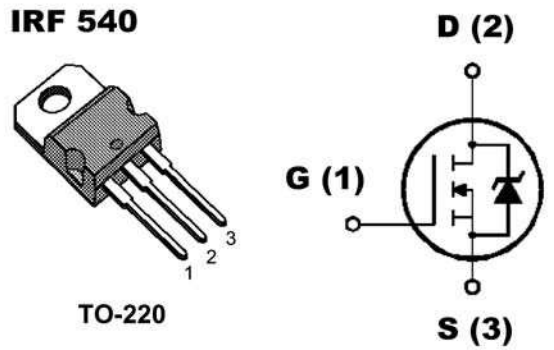


Figura 5 Pinout IRF 540

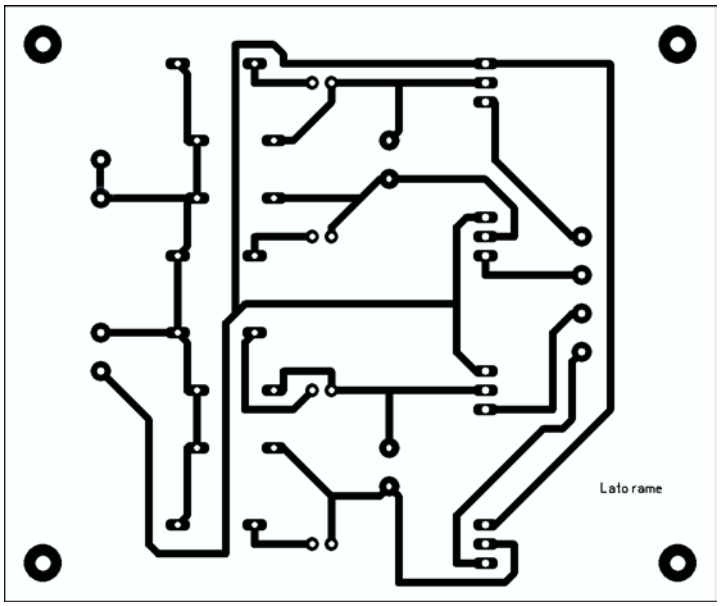


Figura 6 Circuito stampato dell'interfaccia in scala 1:1 (lato rame)

Figura 7 Piano di montaggio dei componenti

Elenco componenti	
Sigla	**Valore**
R1÷R4	470 Ω 1/2 W
D1÷D4	1N4007 o equivalente
LED1÷LED4	Diodo LED
Q1÷Q4	IRF540
X1÷X5	Morsettiera a 2 posti
G1	Morsettiera a 2 posti

Nel saldare i componenti, con un saldatore da 30 W, occorre fare attenzione a non surriscaldarli, inoltre, maneggiate con cura i Mosfet, in quanto una scarica elettrostatica sul terminale Gate potrebbe danneggiarli. Infine prestate cura al montaggio dei diodi, rispettando il catodo (anello bianco per gli 1N4007 e tacca piatta per i diodi led). In figura 8 si può osservare la realizzazione del circuito completo.

I componenti

I componenti utilizzati in questo progetto sono facilmente reperibili. Per quanto riguarda i diodi di protezione, si possono impiegare anche tipi equivalenti, lo stesso dicasi per i mospower. In tabella è presente la lista dettagliata con tutti i componenti elettronici utilizzati nel circuito.

Riconoscimento dei fili di uscita

Quando abbiamo tra le mani un motore passo-passo, la nostra prima reazione sarà probabilmente di ammirazione. Un oggetto di questo tipo è infatti il frutto di alta tecnologia alla portata di tutti. Ma subito dopo sopravviene una sorta di timore: quali saranno i corretti fili di pilotaggio?
I diversi colori causano sempre confusione e il motore è quasi sempre sprovvisto di qualsiasi indicazione o targhetta. Vi proponiamo pertanto un metodo semplice per la determinazione dei fili di uscita e della loro funzionalità. Tale procedimento prende il nome di metodo combinatorio.
Per i nostri esperimenti useremo solamente i motori unipolari, sicuramente più diffusi e soprattutto più semplici da pilotare. Da questi possono uscire 5, 6 oppure 8 fili, ma il procedimento è identico per tutti i modelli. Se i fili sono **5**, uno va collegato all'alimentazione positiva, gli altri quattro costituiscono le bobine. Se i fili sono **6**, due vanno collegati all'alimentazione positiva, gli altri quattro costituiscono le bobine.

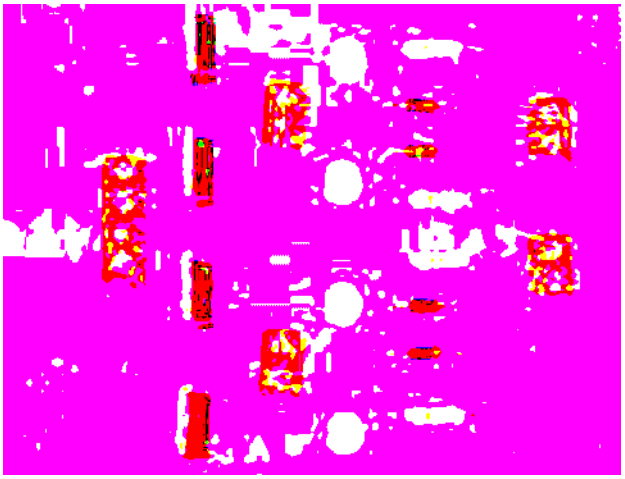


Figura 8 *Il prototipo completo*

Infine, se i fili sono **8**, quattro vanno collegati all'alimentazione positiva, gli altri quattro costituiscono le bobine.
Prendiamo quale esempio un ipotetico motore unipolare, provvisto di 6 fili colorati, ma senza alcuna indicazione di funzionalità, quindi completamente indecifrabile. La figura 9 riporta un tipico motore passo-passo a 6 fili.

In pratica occorre servirsi di un tester (in posizione ohm) misurando, a due a due, tutti i possibili accoppiamenti dei fili. Dal momento che il nostro motore è dotato di sei fili, dovremo effettuare 15 misurazioni diverse. In tabella 1 i risultati ottenuti.

Come potete notare dalle misure effettuate, ci troviamo davanti a bobine di fase della resistenza tipica di 10 ohm cadauna. Analizzando la tabella notiamo subito (vedi misure in grassetto) che solamente due forniscono una resistenza di 20 ohm (il doppio di 10 ohm) e che tale risultato deve esclusivamente addebitarsi al collegamento **in serie** di due bobine. Pertanto, guardando anche la struttura di figura 2, possiamo tranquillamente affermare quanto segue:

- I fili contrassegnati con colore Rosso, Giallo, Marrone e Arancio sono sicuramente quelli che

N.	Colori	Misura
01	Rosso-Nero	Aperto
02	Rosso-Giallo	**20 ohm**
03	Rosso-Bianco	10 ohm
04	Rosso-Marrone	Aperto
05	Rosso-Arancio	Aperto
06	Nero-Giallo	Aperto
07	Nero-Bianco	Aperto
08	Nero-Marrone	10 ohm
09	Nero-Arancio	10 ohm
10	Giallo-Bianco	10 ohm
11	Giallo-Marrone	Aperto
12	Giallo-Arancio	Aperto
13	Bianco-Marrone	Aperto
14	Bianco-Arancio	Aperto
15	Marrone-Arancio	**20 ohm**

Tabella 1 *Risultati ottenuti*

Figura 9 *Un motore passo-passo "indecifrabile"*

devono essere eccitati con un **impulso negativo**, per far procedere il motore di uno step.
- I fili contrassegnati con colore Bianco e Nero vanno collegati al positivo dell'alimentazione.
- Il filo Bianco è connesso elettricamente alle bobine con filo di colore Rosso e Giallo.
- Il filo Nero è connesso elettricamente alle bobine con filo di colore Marrone e Arancio.

Per quanto riguarda invece la corretta sequenza degli impulsi, questa deve essere provata a tentativi, ma dal momento che già conosciamo la funzione di tutti i fili, la maggior parte del lavoro di riconoscimento è concluso. In figura 10 riportiamo lo schema del motore con i fili colorati (e riconosciuti) seguendo il metodo proposto.
Ricordate che i colori dei fili cambiano da motore a motore, pertanto non si può generalizzare una regola, seppur approssimativa, per la determinazione delle funzioni di uscita.

Utilizzo avanzato della scheda

La nostra interfaccia può essere collegata anche ad un PC (oltre che ad un microcontrollore). I relativi ingressi pertanto si collegheranno ai relativi pin della porta parallela, il tutto coadiuvato da un opportuno software scritto in linguaggio Assembler, C o Basic.

Migliorie dell'interfaccia

In ogni caso la scheda di pilotaggio appena realizzata costituisce un prototipo con cui si può facilmente sperimentare. Si potrebbero apportare eventuali migliorie atte ad aumentarne la sicurezza e le prestazioni; in tal senso si potrebbero prevedere i seguenti accorgimenti:

- Aggiungere alcuni buffer in entrata per separare gli stadi ed adattarne meglio le impedenze.
- Optoisolare gli ingressi per prevenire danni accidentali al microcontrollore o al PC.
- Proteggere i mos-power da eventuali blocchi motore o cortocircuito degli induttori, che causerebbero certamente danni ai componenti nonché alle piste del circuito stampato (parlo per esperienza…).
- Prevedere un sistema di correzione di errore nel caso che il motore "perda" qualche step.

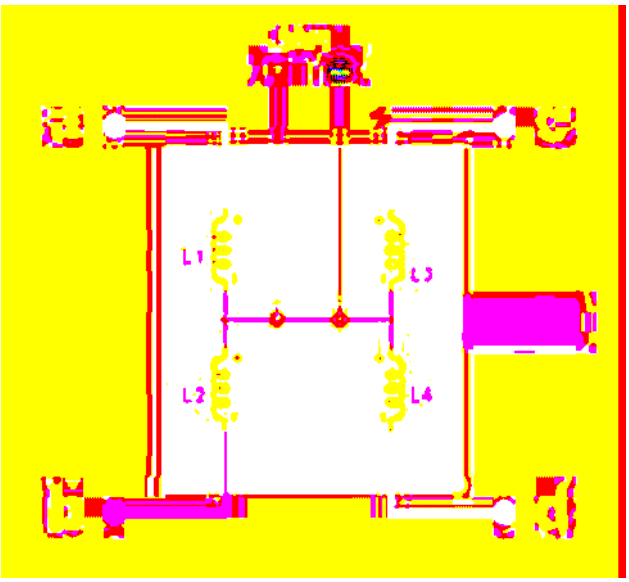


Figura 10 *Schema motore riconosciuto e codificato*

Finalmente mettiamo in moto

Dopo questa lunga ma necessaria premessa passiamo al collegamento dell'interfaccia e del motore al microcontrollore.

Il nostro motore

Per i nostri scopi è stato utilizzato un motore con le seguenti caratteristiche:

- Tensione nominale: 12 Volt.
- Sei fili colorati uscenti.
- Tipo unipolare.
- Colore dei fili collegabili all'alimentazione positiva: bianco, nero.
- Colore dei fili delle quattro fasi: rosso, giallo, marrone, arancio.
- Numero di passi o step: 48.
- Angolo di uno step: 7,5°.
- Numero massimo giri al secondo: 8 RPS.
- Numero massimo STEP al secondo: 384.
- Durata minima di uno step: 3 ms.

Schema di collegamento

Utilizzate lo schema riportato in figura 11. Si tratta di un assemblaggio di quattro parti fondamentali: il microcontrollore, l'interfaccia, il motore e l'alimentazione (diversa per due unità). Montate e collegate correttamente i quattro moduli per poter seguire gli esempi dell'articolo.
Ricordatevi soprattutto di collegare tra loro le

masse dei diversi moduli (scheda PIC, interfaccia ed alimentazione). Il PIC, come sempre, deve essere alimentato con una tensione di 5 Volt.

Un giro e si ferma

Bene, come primo esempio proponiamo un problema semplice da risolvere. L'esercizio è il seguente:

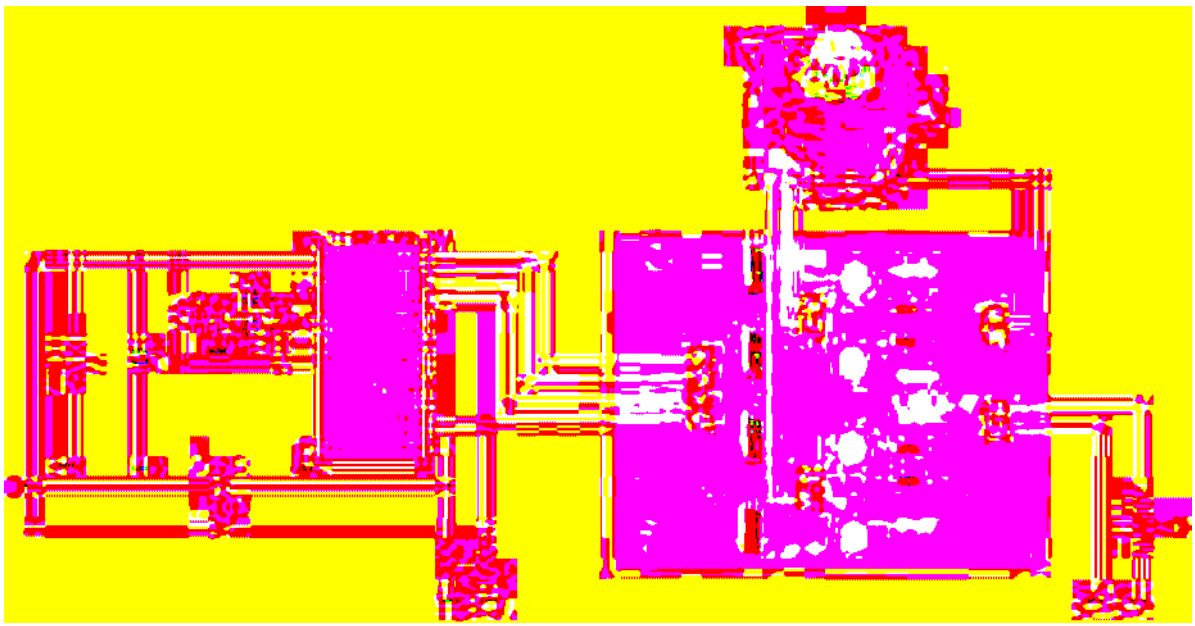


Figura 11 Schema applicativo completo

Software Mikrobasic

Un potente compilatore Basic per PICmicro

- Code Editor
- Code Explorer
- Debugger
- Statistiche

Tutto in un ambiente Windows facile ed intuitivo

Un set di strumenti veramente indispensabili per sviluppare applicazioni con i PICmicro

Ordinalo subito su www.farelettronica.com oppure telefona allo 02.66504794

- All'accensione del circuito c'è una pausa di 2 secondi.
- Quindi il motore compie un giro completo attorno al proprio asse.
- Infine si ferma definitivamente.

Ricordiamo che il motore usato nell'esempio ha una "risoluzione" di 48 step. Esaminiamo il listato 1.

[Listato 1]

```
rem   Dopo una pausa di 2 secondi
rem   il motore compie un giro completo
rem   infine si ferma

program motore01

dim k as byte

portb=0  'Spegne porta B
trisb=0  'Definisce tutta PORTB in OUT

delay_ms(2000)   'Attesa 2 secondi

for k=1 to 12
  portb=1         'Step 1 motore
  delay_ms(100)
  portb=2         'Step 2 motore
  delay_ms(100)
  portb=4         'Step 3 motore
  delay_ms(100)
  portb=8         'Step 4 motore
  delay_ms(100)
next k
portb=0   'Spegne la PORTB

end
```

Il listato è molto semplice. Dopo aver definito la variabile k, che ha la funzione di eseguire tante volte il ciclo di pilotaggio, la PORTB viene azzerata ed inizializzata.
Si passa quindi alla generazione della pausa iniziale attraverso il comando delay_ms. Adesso viene il bello.
Il motore per potere girare ha bisogno di un impulso (a massa) per ogni filo di fase. Impulso che deve avere una durata specifica, nel nostro esempio, di 100 ms. Occorre pertanto "abilitare" (uno per volta) i primi quattro piedini del PIC, per poter "chiudere" i mos-power ed eccitare le bobine.
Ogni qualvolta che si imposta a 1 logico un bit della PORTB, il motore avanza di uno step avanti. Per poter abilitare in sequenza, ed uno per volta, i quattro pin, dobbiamo impostare la PORTB rispettivamente ai valori di 1, 2, 4, 8. Guardiamo la tavola della verità, riportata in tabella 2, per capire bene come avvengono gli impulsi di pilotaggio.

Cosa succede all'interno del ciclo FOR-NEXT? Le quattro istruzioni di assegnazione dei valori alla PORTB fanno semplicemente avanzare il motore di 4 step, pertanto se esso deve compiere un giro completo (48 steps del nostro stepper) occorre ripetere l'intera sequenza per 12 volte (48 : 4 = 12).
Infine, all'uscita del ciclo troviamo l'istruzione che azzera la PORTB, permettendo al motore di fermarsi e di interrompere la corrente alle bobine.

La retromarcia

Se volessimo far avanzare il motore in senso

# Step	Valore decimale PORTB	RB0	RB1	RB2	RB3
1	**1**	**1**	0	0	0
2	**2**	0	**1**	0	0
3	**4**	0	0	**1**	0
4	**8**	0	0	0	**1**

Tabella 2 *Tavola della verità per il pilotaggio a passo intero (marcia avanti)*

# Step	Valore decimale PORTB	RB0	RB1	RB2	RB3
1	**8**	0	0	0	**1**
2	**4**	0	0	**1**	0
3	**2**	0	**1**	0	0
4	**1**	**1**	0	0	0

Tabella 3 *Tavola della verità per il pilotaggio a passo intero (marcia indietro)*

opposto, per nostra fortuna non occorre invertirne la polarità, come avviene per i normali motori, bisogna invece "ribaltare" la sequenza degli impulsi da cadenzare al motore. Pertanto i valori decimali da fornire alla scheda saranno 8, 4, 2, 1 anziché 1, 2, 4, 8. In tabella 3 riportiamo la tavola della verità per la marcia indietro. Provate da voi lo stesso esercizio del listato 1, cambiando solamente il senso di marcia.

E adesso si corre

A differenza dei normali motori, per aumentare o diminuire la velocità di rotazione di un motore passo-passo è sufficiente variare il tempo tra un impulso e il successivo.
Naturalmente non si deve utilizzare un tempo di attesa molto breve, in quanto il motore farebbe "fatica" a seguirlo, con la conseguente perdita di qualche step o addirittura del mancato funzionamento. Consultate a tale proposito le caratteristiche intrinseche del motore.
Si vedrà nel prossimo esercizio come, variando semplicemente il tempo di attesa tra un passo e l'altro, il motore acquisterà una maggiore velocità. Strutturiamo pertanto il problema come segue:

- All'accensione del circuito c'è una pausa di attesa di 2 secondi.
- Il motore compie 5 giri completi a velocità bassa.
- Il motore compie 5 giri completi a velocità medio-bassa.
- Il motore compie 5 giri completi a velocità media.
- Il motore compie 5 giri completi a velocità medio-alta.
- Il motore compie 5 giri completi a velocità alta.
- Infine si ferma definitivamente.

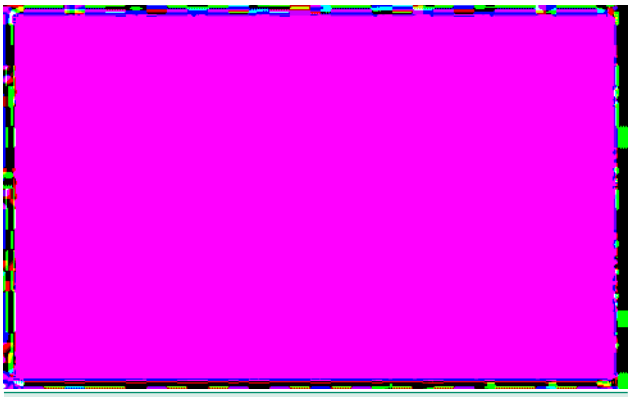


Figura 12 *Analisi uscite del microcontrollore (pilotaggio a passo intero, marcia avanti)*

I termini velocità bassa, media e alta per il momento sono puramente indicativi.
Nel proseguo degli esercizi affronteremo come calcolarli.

[Listato 2]

```
rem   Dopo una pausa di 2 secondi
rem   il motore modifica la propria
rem   velocità di rotazione

program motore02

dim k as byte

portb=0  'Spegne porta B
trisb=0  'Definisce la PORTB in OUT

delay_ms(2000)   'Attesa 2 secondi
'---Velocità bassa---
for k=1 to 60       '5 giri
  portb=1         'Step 1
  delay_ms(100)
  portb=2         'Step 2
  delay_ms(100)
  portb=4         'Step 3
  delay_ms(100)
  portb=8         'Step 4
  delay_ms(100)
next k
'---Velocità medio-bassa---
for k=1 to 60       '5 giri
  portb=1         'Step 1
  delay_ms(80)
  portb=2         'Step 2
  delay_ms(80)
  portb=4         'Step 3
  delay_ms(80)
  portb=8         'Step 4
  delay_ms(80)
```

```
next k
'---Velocità media---
for k=1 to 60       '5 giri
  portb=1         'Step 1
  delay_ms(60)
  portb=2         'Step 2
  delay_ms(60)
  portb=4         'Step 3
  delay_ms(60)
  portb=8         'Step 4
  delay_ms(60)
next k
'---Velocità medio-alta---
for k=1 to 60       '5 giri
  portb=1         'Step 1
  delay_ms(40)
  portb=2         'Step 2
  delay_ms(40)
  portb=4         'Step 3
  delay_ms(40)
  portb=8         'Step 4
  delay_ms(40)
next k
'---Velocità alta---
for k=1 to 60       '5 giri
  portb=1         'Step 1
  delay_ms(20)
  portb=2         'Step 2
  delay_ms(20)
  portb=4         'Step 3
  delay_ms(20)
  portb=8         'Step 4
  delay_ms(20)
next k

portb=0   'Spegne la PORTB

end.
```

Eseguite il programma e noterete che ogni 5 giri il motore subisce un'accelerazione netta. Esaminate bene il listato ed eseguite tutte le variazioni e modifiche di sorta, per provare come si comporta il vostro stepper alle diverse velocità e sensi di marcia.
Una piccola annotazione: poiché, durante la rotazione, è alquanto difficoltoso contare il numero dei giri del motore, consigliamo di agganciare una molletta da bucato sull'albero, come se fosse una lancetta di orologio. Vedrete che il controllo della rotazione sarà sicuramente più agevole.

Il mezzo passo

Il mezzo passo (*half step*), è una tecnica che serve ad aumentare la risoluzione del motore. In altre parole, applicando tale metodo al nostro motore (da 48 step) ne otteniamo uno da 96 passi.
Raddoppia così la precisione dei movimenti a discapito naturalmente della velocità e della coppia motrice.
Vediamo in cosa consiste. In pratica occorre eccitare le varie fasi alternando l'erogazione di corrente ora ad una bobina, ora a due bobine. La tavola della verità di cui alla tabella 4 chiarirà ogni dubbio.

Nel listato 3 è riportato il programma per ottenere la rotazione del motore di un giro completo, sfruttando proprio il mezzo passo. Utilizzando tale metodo c'è un prezzo da pagare: la velocità massima del motore risulta dimezzata ed, inoltre, la coppia dello stesso varia in modo anomalo poiché si alternano due diverse fasi di eccitazione delle bobine in modo misto.

# Step	Valore decimale PORTB	RB0	RB1	RB2	RB3
1	1	1	0	0	0
2	3	1	1	0	0
3	2	0	1	0	0
4	6	0	1	1	0
5	4	0	0	1	0
6	12	0	0	1	1
7	8	0	0	0	1
8	9	1	0	0	1

Tabella 4 *Tavola della verità per il pilotaggio a mezzo passo (marcia avanti)*

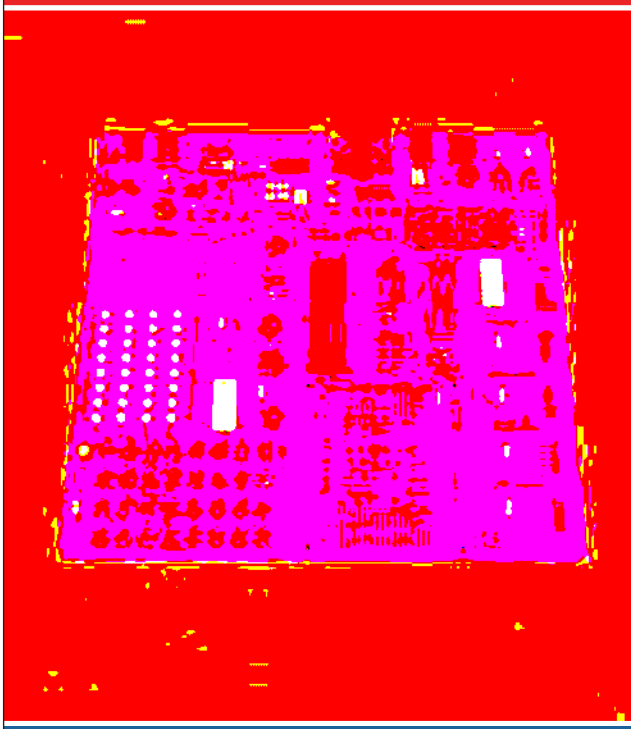
[Listato 3]

```
rem   Il mezzo passo o HALF STEP
program motore03

dim k as byte

portb=0   'Spegne la PORTB
trisb=0   'Definisce la PORTB in OUT

delay_ms(2000)   'Attesa 2 secondi

for k=1 to 12   '1 giro
  portb=1
  delay_ms(100)
  portb=3
  delay_ms(100)
  portb=2
  delay_ms(100)
  portb=6
  delay_ms(100)
  portb=4
  delay_ms(100)
  portb=12
  delay_ms(100)
  portb=8
  delay_ms(100)
  portb=9
  delay_ms(100)
next k

portb=0   'Spegne la PORTB

end.
```

Esempio finale

Come esercizio finale proponiamo il seguente problema: un motore stepper gira inizialmente in senso orario. La velocità poco importa. Se l'utente preme il pulsante posto sul pin RA1 del PIC, il motore inverte la marcia, mentre se preme il pulsante posto su RA0 viene ripristinato il senso di rotazione iniziale. Riferiamoci sempre allo schema di cui in figura 11. Digitiamo il listato 4 e passiamo ai commenti di rito.

Scheda easyPIC3



La rivoluzionaria scheda di sviluppo per PICmicro

- Programmatore USB2.0 on-board
- Tastiera a 32 tasti
- 32 LED per il monitoraggio degli I/O
- 4 cifre LED a 7 segmenti
- Predisposizione per moduli LCD alfanumerici
- Predisposizione per moduli LCD grafici
- Predisposizione per comunicazione RS232
- Predisposizione per tastiera PS2
- Predisposizione per sensore di temperatura DS1820
- Supporto per tutte le famiglie PIC (anche PIC10F)
- Predisposizione per comunicazione USB
- Alimentazione esterna o via USB
- Fornita con 16F877
- Disponibile con o senza display

Ordinala subito su www.farelettronica.com oppure telefona allo 02.66504794

[Listato 4]

```
rem Inizialmente il motore va in senso orario.
rem Se si preme il tasto su RB1 cambia il senso
rem mentre se si preme il tasto su RB0
rem si ripristina il senso iniziale

program motore04
dim marcia as byte  '(0=avanti  1=indietro)

porta=0   'Azzera PORTA
porta=%00011  '(bit 0 e 1 in Input)

portb=0   'Spegne porta B
trisb=0   'Definisce la PORTB in OUT

marcia=0   'Inizialmente va avanti

while true
  '---Processa i tasti---
  if porta.1=1 then       'Se premo su RA1
    marcia=1
  end if
  if porta.0=1 then       'Se premo su RA0
    marcia=0
  end if
  '---Va in senso orario---
  if marcia=0 then
    portb=1         'Step 1
    delay_ms(50)
    portb=2         'Step 2
    delay_ms(50)
    portb=4         'Step 3
    delay_ms(50)
    portb=8         'Step 4
    delay_ms(50)
  end if
  '---Va in senso antiorario---
  if marcia=1 then
    portb=8         'Step 1
    delay_ms(50)
    portb=4         'Step 2
    delay_ms(50)
    portb=2         'Step 3
    delay_ms(50)
    portb=1         'Step 4
    delay_ms(50)
  end if
wend

end.
```

L'intero algoritmo è imperniato sulla variabile *marcia*, utilizzata come indicatore di stato (flag). Quando i tasti sono processati e valutati, è proprio questa variabile ad assumere permanentemente (e non solo durante la pressione dei tasti) il valore di 0 oppure 1.
Pertanto il flag viene poi analizzato dalle rispettive istruzioni di condizione *IF* e di conseguenza eseguiti i relativi statement di marcia avanti e marcia indietro.

RACCOMANDAZIONI FINALI

L'uso dei motori stepper non è per nulla pericoloso, ma costituisce un primo approccio all'elettronica di potenza, in quanto la corrente utilizzata nei circuiti comincia a crescere. Pertanto, durante la progettazione e l'utilizzo di questi tipi di motore consigliamo di prendere alcune precauzioni:

- Controllare sempre la tensione nominale massima sopportata dal motore.
- Verificare il numero di step del motore.
- Non alzare troppo la velocità: il motore perderebbe qualche "colpo" e si ridurrebbe la coppia.
- Tutti i motori generano un'elevata tensione di ricircolo, dovuta agli induttori; prevedere pertanto dei diodi veloci in antiparallelo.
- Se i mos-power dovessero riscaldare, applicare delle alette di alluminio adeguate per la dispersione di calore.
- Se, utilizzando un alimentatore, il motore dovesse comportarsi in maniera "strana" rispetto le vostre previsioni, inserire un grosso condensatore elettrolitico (almeno 10.000 microfarad) in parallelo all'alimentazione; evidentemente si sono verificati piccoli "ammanchi" di corrente dovuti allo *spunto* del motore. Il condensatore funge in questo caso da erogatore, durante i tempi morti.
- Agganciare una molletta da bucato all'albero rotante: potrete così contare facilmente i singoli passi e i giri del motore.

CONCLUSIONI

Come abbiamo visto, un altro importante tassello è stato inserito nello studio di questo fantastico compilatore che è il Mikrobasic. A presto dunque con tanti altri interessanti articoli di approfondimento. Buona sperimentazione a tutti.

TELERADIO & COLLEZIONI

PIACENZAEXPO

PIACENZA
10-11 DICEMBRE
2005

RADIANTISMO
TELECOMUNICAZIONI
COMPUTER ○ TV - SAT ○ TELEFONIA
ACCESSORI
ed inoltre
LIBRI ○ DISCHI ○ FUMETTI
GIOCATTOLI ○ MINERALI
COLLEZIONISMO

nel periodo ideale per gli acquisti ed i regali

PIACENZAEXPO
10-11 DICEMBRE 2005

s.s. 10 fraz. Le Mose - 29100 Piacenza
tel. 0523 602711 - fax 0523 602702
e-mail: info@piacenzaexpo.it

USCITE AUTOSTRADALI

ORARI

WWW.TELERADIO.PIACENZAEXPO.IT

gli appuntamenti

05-06 Novembre 2005

SALONE DELL'ELETTRONICA — Erba (CO)

EXPO ELETTRONICA (ex ABC dell'Elettronica) ad Erba (Como) si svolge due volte all'anno, in primavera ed in autunno. Oltre alle merceologie "tradizionali" proposte da questo tipo di manifestazioni, quali computer, elettronica in genere, radiantismo, telefonia, surplus... nonché radio d'epoca, dischi e CD da collezione.
Inoltre propone, in primavera, il Salone dell'Astronomia e Photo Cine video, con macchine ed attrezzature per la fotografia , mentre in autunno i protagonisti sono i radioamatori grazie al CB Day, a loro dedicato.
Certamente non mancano i buoni motivi per andare a dare un'occhiata; ci saranno buoni affari sia per gli esperti sia per i neofiti!

Luogo: Centro Espositivo e Congressuale Lario Fiere Erba (CO)
Orari: Dalle 9:00 alle 18:00
Organizzatore: Blu Nautilus srl (www.blunautilus.it - Tel 0541.439573)
Ingresso: € 7,50 (ridotto € 6,50)

13-14 Novembre 2005

33° ELETTROEXPO — Verona

Mostra mercato di materiale radiantistico per CB e radioamatori, apparecchiature per telecomunicazioni, telefonia, antenne e parabole per radioamatori e TV Sat, radio d'epoca, stampa specializzata.

Luogo: Verona Fiere - Verona
Orari: Dalle 9:00 alle 19:00
Organizzatore: Verona Fiere (www.veronafiere.it - Tel 045.8298111)
Ingresso: € 8,00

19-20 Novembre 2005

COMPUTERFEST & RADIOAMATORE — Reggio Emilia

Nell'arco dei due giorni di durata della fiera, sarà possibile osservare, confrontare, ed acquistare singoli componenti, periferiche, software e personal computer; apparati radio e componentistica radioamatoriale; home entertainment (TV-Sat, DVD, antenne); telefonia fissa e mobile; editoria specializzata; fai-da-te elettronico, attrezzature e strumentazioni varie. Per tutti i visitatori dell'edizione estiva di Cerea, ancora muniti di biglietto, l'ingresso sarà completamente gratuito.

Luogo: Fiere Di Reggio Emilia - Reggio Emilia
Orari: Dalle 9:00 alle 18:00
Organizzatore: Compendio Fiere S.r.l. (www.compendiofiere.it - Tel. 337.676719)
Ingresso: € 7,00 (ridotto € 5,00)

di NOVEMBRE

duemilacinque

Per sapere a quali appuntamenti sarà presente visita la pagina: www.farelettronica.com/fiere

24 Novembre 2005

ITF 2005 — Torino

L'evento, alla sua quarta edizione, è l'unica iniziativa a livello nazionale in grado di collocarsi come punto di riferimento per il mondo della telematica e dell'infomobilità, fotografando un mercato complesso e con dinamiche molto vivaci.
Reti veicolari wireless, localizzazione e gestione flotte per l'autotrasporto professionale, RFID, GIS e GPS, applicazioni per il trasporto pubblico, le forze dell'ordine, la protezione civile e le emergenze sanitarie sono stati i temi caldi già affrontati, anticipando quanto avverrà sul mercato italiano e internazionale nel 2006, che verranno riproposti e affiancati da nuovi focus di interesse.
La quarta edizione di ITF verrà ospitata, insieme al VETIS, all'interno dell'AUTOMOTIVE WEEK, che ha lo scopo di promuovere nuovi contatti professionali tra compratori e fornitori del settore automobilistico.

Luogo: Centro Congressi Lingotto - Torino
Orari: Dalle 9:00 alle 19:00
Organizzatore: Wireless S.r.l. (www.wirelessforum.it)
Ingresso: Gratuito (riservato agli operatori del settore)

26-27 Novembre 2005

7° RADIOAMATORE 2 — Pordenone

Mostra mercato di apparecchi radio, componenti e ricambi, attrezzature ed accessori, computer e accessori, videogiochi, software.

Luogo: Quartiere Fieristico Pordenone - Pordenone
Orari: Dalle 9:00 alle 18:00
Organizzatore: Pordenone Fiere. (www.fierapordenone.it - Tel. 0434.232111)
Ingresso: n.p.

03-04 Dicembre 2005

GRANDE FIERA DELL' ELETTRONICA — Forlì (FC)

Alla Fiera di Forlì appuntamento con uno dei più importanti eventi nazionali "consumer" di settore; quello con la Grande Fiera dell'Elettronica. Le cifre sono quelle delle grandi occasioni: oltre 350 espositori, 20.000 mq di esposizione e più di 20.000 visitatori attesi alla manifestazione. La Grande Fiera dell'Elettronica ha un pubblico vasto ed eterogeneo: appassionati del "fai da te", elettro-riparatori, "smanettoni", radioamatori, "cacciatori" di buone occasioni o pezzi rari; questo perché propone un panorama merceologico e un calendario di eventi collaterali veramente ricchissimo. Al suo interno si svolgono diversi saloni tematici: La Fiera dell'Astronomia amatoriale, Photo cine video, Radio Expo mercatino delle radiocomunicazioni, Dischi e cd da collezione e il concorso nazionale dell'inventore elettrico-elettronico.

Luogo: Fiera di Forlì - Forlì (FC)
Orari: Dalle 9:00 alle 18:00
Organizzatore: Blu Nautilus srl (www.blunautilus.it - Tel 0541.439573)
Ingresso: € 8,00 (ridotto € 7,00)

Il progetto LIRC

In questo articolo ci occuperemo della costruzione di un'interfaccia infrarossi per comandare una qualsiasi applicazione da remoto, utilizzando un dispositivo IR come un telecomando. La sequenza di argomenti parte dalla presentazione del progetto software Lirc, passa attraverso la realizzazione dell'interfaccia ed arriva ai possibili utilizzi con la configurazione dei diversi software disponibili a corredo dell'hardware autocostruito

COS'È LIRC?

LIRC (**L**inux **I**nfrared **R**emote **C**ontrol) è un progetto che nasce per permettere di decodificare ed inviare segnali infrarossi realizzando connettività tra PC e i più comuni telecomandi disponibili o tra un PC e i vari dispositivi che possono essere comandati da remoto tramite IR. Nel nostro caso ci occuperemo del primo tipo di utilizzo, cioè cercheremo di capire come costruire un ricevitore per poter comandare il PC con un qualsiasi telecomando che è possibile trovare in ambito domestico.
Esistono diversi tipi di ricevitori che è possibile realizzare da utilizzare con la parte software del progetto, i cui schemi sono proposti sul sito *www.lirc.org*. Tali ricevitori differiscono principalmente per il collegamento con il PC, esistono ricevitori da collegare alla porta seriale o alla porta parallela, infine anche alla porta USB (vedi figura 1,3,4 tratte dal sito lirc.org), sceglieremo tra queste realizzazioni quella più economica e più semplice così da passare subito all'azione.
In origine il progetto si basava solo su ricevitori autocostruiti mentre con il susseguirsi delle versioni disponibili l'hardware supportato è aumentato ed è stato allargato anche a quello commerciale. La versione corrente supporta un'ampia varietà di hardware, come i ricevitori: *Irman, Anir Multimedia Magic, Packard Bell, PixelView RemoteMaster, Logitech/AST*, le porte *IrDA* in modalità seriale disponibili nei notebook, i ricevitori *Pinnacle Systems PCTV* (pro) e l'*Slink-e* di *Nirvis Systems*.
È già stata realizzata l'estensione del progetto al completo supporto delle porte IrDA diffuse sui computer portatili. In generale quasi tutti i telecomandi dei normali apparecchi domestici funzionano con LIRC, tranne quelli che per scelte tecnologiche molto complicate risultano oggettivamente fuori dalla portata dei normali ricevitori, è possibile trovare una lista estesa ma non esaustiva all'indirizzo: *lirc.sourceforge.net/remotes*
Veniamo quindi a descrivere la parte software del progetto così da indicarne le potenzialità. Il tutto si basa sul demone *lircd*, fondamentale per il funzionamento del ricevitore, in quanto è un processo sempre attivo in attesa di richieste provenienti dall'interfaccia a cui è collegato il ricevitore. Poi abbiamo un secondo demone chiamato *lircmd*, che comunica con *lircd*, che si occupa di tradurre i comandi inviati all'interfaccia e decodificati da *lircd* in comandi del mouse o della tastiera. Lo spazio delle applicazioni di utilizzo permette di con-

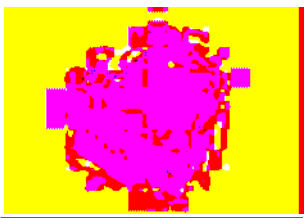


Figura 1 Ricevitore Seriale

Figura 2 Logo LIRC

Realizzazione di un'interfaccia IR-PC

di Enrico Raffone
enrico.raffone@tin.it

trollare un qualsiasi computer attraverso un telecomando, si può per esempio arrivare a realizzare un lettore DVD/DIVX con un vecchio PC ed un telecomando.

COS'È WINLIRC?

LIRC è un progetto nato in ambito Linux, **WinLIRC** è semplicemente il porting di tale progetto in ambiente Windows. Per la maggiore diffusione del sistema operativo di casa Microsoft si è scelto di indicare nella sezione di configurazione del software a complemento del ricevitore, soprattutto applicazioni che si basano su WinLIRC.

COSTRUIRE L'INTERFACCIA INFRAROSSI

Sui siti indicati e su quello di WinLIRC (winlirc.sourceforge.net) sono indicati diversi schemi, tra questi scegliamo il più economico per il numero di componenti ed il più semplice in quanto non necessita di alcun tipo di microcontrollore, consideriamo quindi un ricevitore da collegare alla porta seriale del PC secondo lo schema di figura 5.
Nell'elenco componenti sono indicati tutti i principali, l'ultimo dei quali è il ricevitore infrarossi. Fulcro del dispositivo che si va realizzare, tale ricevitore dovrà essere capace di ricevere segnali modulati a 38 KHz come accade nella maggior parte dei telecomandi per dispositivi ad uso casalingo. Esistono diversi costruttori che propongono tale tipo di ricevitori con una diversa sensibilità ai disturbi, alleghiamo una lista abbastanza completa, dove sono indicati i ricevitori compatibili per la nostra applicazione. Nella realizzazione qui descritta utilizzeremo i ricevitori TSOP 1138 della Vishay Telefunken, indicati per questo tipo di applicazione tra i migliori:

- **Vishay** TSOP 1738 o 1838 o 1138
- **Siemens** SFH 5110
- **Radio Shack** 276-0137
- **Everlight** IRM 2238 o 8100-3-M o 8601M o 8601S
- **Mitsumi IR Preamp** KEY-COOSV (0924G)
- **Toshiba** TK19 444 TFMS 5360 (conosciuto in Italia come Mivar IC)
- **Temic** TFMS 5380
- **Sony** SBX 1404-01 o 1620-12
- **Sharp** 1u521X o GP1U52XB o GP1U271R o GP1UD261XK
- **Panasonic** PNA4602M o PNA4611M o PNA4612M

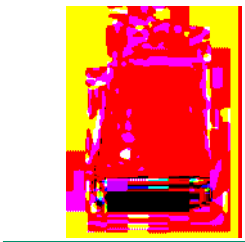


Figura 3 Ricevitore parallelo

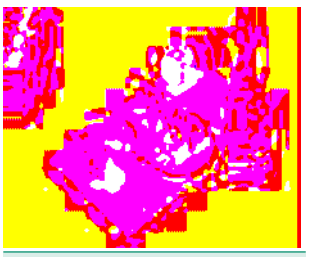


Figura 4 Ricevitore USB

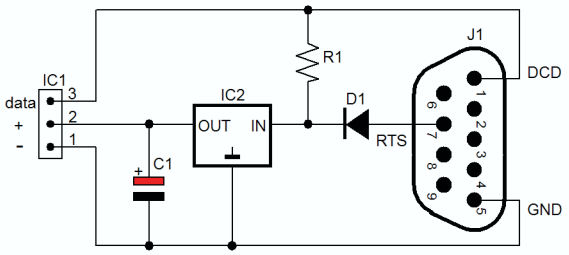


Figura 5 Schema ricevitore seriale

Elenco componenti	
Sigla	**Valore**
J1	Connettore Dsub9 femmina
D1	1N4148
R1	4.7 KΩ 1/4 W
IC2	LM7805
C1	4.7 µF 16 V elettrolitico
IC1	Ricevitore IR (vedi lista)

- **Kodenshi** PIC-12043S
- **Daewoo** DHR-38 C 28
- **A-Tronic** TRM38 IR module

Per quanto riguarda la reperibilità dei ricevitori, c'è un serio problema, soprattutto se non si vuole far riferimento alle possibilità di acquisto online di componenti elettronici. Questo tipo di ricevitori non si trova facilmente nei negozi di elettronica, in quanto risulta più un componente di sostituzione per riparatori TV. In ogni caso dopo diverse ricerche è stato possibile reperire un Vishay TSOP 1138 alla cifra di € 3,20. Si consiglia vivamente di cercare tali ricevitori su siti di vendita online di prodotti elettronici o interpellando un produttore, come per esempio la Vishay, che si è dimostrata molto gentile e disponibile nei confronti dei clienti ed in particolare dello scrivente.

Figura 6 Vishay TSOP 1838 e 1138

Figura 7 Vishay TSOP 1838 e 1138

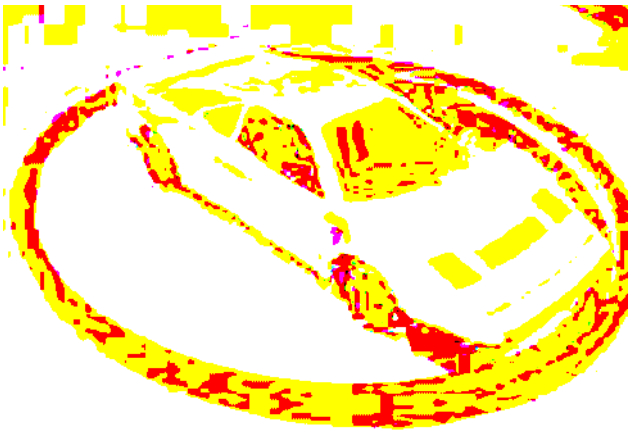


Figura 8 Realizzazione ricevitore in un modellino scala 1:25

REALIZZAZIONE DEL RICEVITORE IN UN MODELLINO

Per la realizzazione del ricevitore al fine di migliorare l'estetica, si è scelto di inserire il circuito all'interno di un modellino di una *Ferrari GTO* scala 1:25. Reperito un cavo da due poli più schermo lungo circa un metro, si è realizzato il circuito su di un pezzo di millefori ed inserito nell'abitacolo del modellino, praticando un foro sul retro attraverso cui passasse il cavo che giunge alla porta seriale.
Iniziamo col chiarire i collegamenti con il connettore DSub9, bisogna individuare i pin 1, 5, 7 come indicato in figura 5, il connettore è schematizzato in figura 9, in cui sono riportati anche i collegamenti.
I componenti sono stati disposti su millefori secondo lo schema in figura 10. ed inseriti nell'abitacolo del modellino (figura 11).

INSTALLAZIONE E CONFIGURAZIONE DEL SOFTWARE

L'interfaccia appena costruita è utilizzabile sia in ambiente Linux, che in quello Windows, ma, come già anticipato nei precedenti paragrafi, prenderemo come riferimento il sistema operativo Microsoft, in quanto in questa sede si vuole tener conto della maggiore diffusione dell'OS della software house di Redmond.
Quindi considereremo l'utilizzo del porting in Windows del progetto LIRC, WinLIRC, giunto ormai alla versione 0.6.5 e scaricabile liberamente dal sito *winlirc.sourceforge.net.* Oltre a WinLIRC, abbiamo bisogno di un altro software che lavori in collaborazione e che traduca i segnali provenien-

Figura 9 Schema pin DSub9/Femmina

circuiti stampati in 24 ore

garantiamo il tempo di consegna anche i circuiti sono gratis

millennium
mc
data ware

www.mdsrl.it

te dall'interfaccia in comandi per applicazioni. Esistono diversi applicativi che supportano WinLIRC, tra questi:

- **Jlirc** (jlirc.sourceforge.net), Java API per LIRC e WinLIRC.
- **PC Remote Control** (shareware, www.pcremotecontrol.com) supporta ricevitori WinLIRC-compatibili e altri.
- **IReX** (www.ramscan.com/irex) permette di lanciare programmi attraverso un telecomando.
- **Universal Infrared Control Engine** (shareware, www.mediatexx.com) programma che supporta completamente WinLIRC.
- **IRAssistant** (shareware, www.irassistant.com) permette di emulare le azioni del mouse, lanciare applicazioni, eseguire macro ed altro.
- **Odtwarzacz Filmow** (www.koala.pl/). Un Film Player che supporta WinLIRC.
- **BSPlayer** (www.bsplayer.org) un DivX-player freeware che supporta WinLIRC.
- **Girder** (shareware, www.girder.nl) permette di fare di tutto con un telecomando e Windows, supporta WinLIRC attraverso un plugin.
- **Remote Control Plugin for Winamp** (remotectrl.narod.ru). Un Plugin per comandare WinAmp a distanza.
- **Light Alloy** (www.softella.com)un multimedia-player che supporta WinLIRC.
- **MyAlbum** (myalbumpro.com/MyAlbum.html). Un programma per realizzare database di immagini e video che supporta WinLIRC per visualizzare le foto come slides.
- **Light Control** (stephenhaas.tripod.com/vblightcontrol/index.html). Applicazione legata alla domotica, utilizza otto bit della porta parallela per controllare luci e supporta WinLIRC.
- **FlexRemote** (freeware, www.flexremote.com). Programma freeware simile a Girder, che supporta WinLIRC.

Si tiene sottolineare che le scelte che ci hanno guidato nella selezione del software da utilizzare sono legate alla possibilità di disporre di prodotti freeware, quindi liberi da ogni tipo di costo per l'utilizzatore finale, naturalmente per WinLIRC il problema non si pone, mentre per quanto riguarda gli applicativi a supporto di WinLIRC non è sempre possibile usufruire di prodotti free, per esempio molti dei programmi presenti nella lista precedente sono shareware, quindi sono stati scartati nella nostra analisi. Una nota particolare va a Girder, che nasce come programma completamente gratuito per uso privato, uno dei primi programmi se non il primo ad essere utilizzato per questo tipo di applicazione, che di recente è divenuto shareware ed acquisito dalla Promixis.
Una possibilità interessante risulterebbe reperire ed utilizzare una delle ultime versioni completamente free la 3.3.1b, che risulta ancora disponibile nei meandri di Internet.
Tra i diversi software elencati scegliamo per la nostra piccola guida al funzionamento l'ultimo, FlexRemote, programma semplice nell'utilizzo, ma già abbastanza evoluto per l'automazione di Windows.

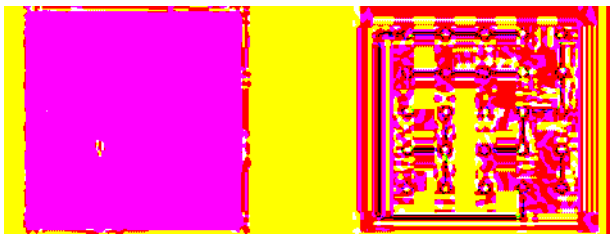


Figura 10 Circuito stampato lato rame e componenti, disegnato per poter facilmente essere realizzato su millefori

Figura 11 Particolare abitacolo con ricevitore IR e condensatore

Installazione e configurazione di WinLIRC 0.6.5

Andiamo al sito *winlirc.sourceforge.net* e scarichiamo il programma. Il download consta di programma eseguibile e codice sorgente che può essere modificato e ricompilato (utilizzando Microsoft Visual C++ 6.0). Ricordiamo che il programma è completamente gratuito ed open source. Viene distribuito con licenza **GPL** (General Public License, della Free Software Foundation *www.fsf.org*). Il programma scaricato è contenuto in un archivio zip, che va decompresso in una

directory. Possiamo decomprimere l'archivio in *C:\Programmi\WinLIRC*, eventualmente è possibile creare un collegamento nel menù programmi e nel caso si voglia usufruire della possibilità di comandare in qualsiasi momento il PC con la nostra interfaccia IR, possiamo anche inserire un collegamento nel gruppo del menù programmi "Esecuzione Automatica", così da caricare il programma all'avvio.

Avviamo il programma da C:\Programmi\WinLIRC, il file di avvio è l'eseguibile *winlirc.exe*. Alla prima esecuzione abbiamo un avviso di errore di inizializzazione (figura 12), cioè è dovuto semplicemente al fatto che non c'è alcun file di configurazione, quindi l'hardware non può funzionare correttamente. Bisogna scegliere di cambiare la configurazione, così da avviare l'impostazione dei parametri dell'hardware. Si passa alla prima schermata di configurazione (figura 13).

Circa i parametri principali da impostare, abbiamo la porta seriale a cui abbiamo connesso l'interfaccia, nel nostro caso è impostata la COM1, lasciamo la velocità della porta a 115200 bps, lasciamo "Sense" in modalità "Autodetect", circa le altre voci c'è da lasciare abilitato come in figura la voce "animax (enable DTR)" che impone l'alimentazione del ricevitore attraverso la porta seriale. Ultima cosa da definire è il nome del file di configurazione, che conterrà i codici relativi ai diversi tasti del telecomando a nostra disposizione, nel caso in figura si è scelto "mio.cf".

Dopo questa prima configurazione c'è una fase di apprendimento, in cui il programma deve acquisire tutti i codici relativi ai diversi tasti disponibili per il nostro telecomando.

Andiamo, quindi, alla fase di apprendimento premendo "Learn".

Figura 12 *La prima schermata di WinLIRC con segnalazione di errore*

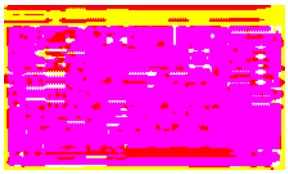


Figura 13 *Configurazione*

Figura 14 *Apprendimento codici telecomando (1)*

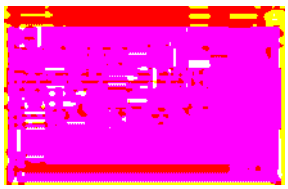


Figura 15 *Apprendimento codici telecomando (2)*

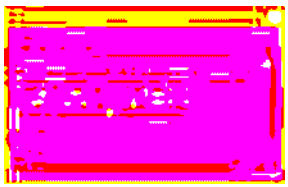


Figura 16 *Apprendimento codici telecomando (3)*

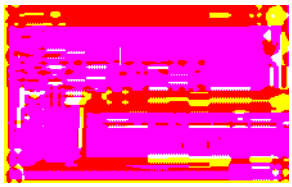


Figura 17 *Apprendimento codici telecomando (4)*

Inseriamo il nome del telecomando a disposizione, tale indicazione può essere utile per inserire il file di configurazione, completato delle informazioni relative al protocollo di trasmissione utilizzato, che otterremo alla fine della fase di apprendimento, nel database dei telecomandi supportati (www.lirc.org/remotes).

Quindi passiamo ad un altro parametro da impostare. Viene indicata la possibilità di errori nell'apprendimento dei codici, a causa di disturbi nella trasmissione della luce infrarossi o a causa di possibili altre imperfezioni nella trasmissione, con un margine di errore che viene indicato in percentuale. Il valore medio intorno al quale si assestano questi errori è il 25%, quindi si richiede di impostare un determinato margine. Si consiglia di premere direttamente invio così da indicare il margine di default a 25%. La successiva schermata (figura 15) richiede all'utente se conosce due parametri relativi al telecomando da utilizzare, cioè gap e signal lenght, ovvero l'intervallo tra due fronti d'onda e la lunghezza del segnale, ma siamo al primo apprendimento, quindi premendo invio verrà richiesta più volte la pressione a caso di pulsanti così da poter rilevare i due parametri automaticamente.

Andiamo al secondo step della fase di apprendimento, vengono ad essere memorizzati i veri e propri codici del telecomando, quindi una fase abbastanza ripetitiva, in cui il programma chiede il nome del tasto e la corrispondente pressione dello stesso, come mostrato in figura 16.

La pressione deve essere continua fino a quando si completa la singola acquisizione del codice relativo al tasto (figura 17). Si passa al prossi-

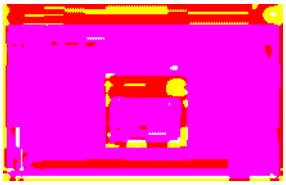


Figura 18 Analisi riuscita

mo tasto come in figura 16 e così via fino a completare l'intero insieme di tasti. A seguito dell'ultimo tasto viene fatta nuovamente la richiesta di memorizzazione di un altro, per completare basta non inserire nulla e premere invio, a questo punto l'inizializzazione è completa.
Il programma consiglia per migliorare le performance di premere il pulsante Analyze (figura 18), a cui corrisponde un confronto dei codici dei diversi pulsanti cercando similitudini tra questi, in modo da ottimizzare il riconoscimento. Il successo dell'ottimizzazione viene indicato come in figura 18, le conseguenze sono la risistemazione del file di configurazione. In caso di insuccesso, il programma indica che l'ottimizzazione non è riuscita e che il telecomando funzionerà in "raw mode", cioè ad ogni comando il suo specifico pattern.
A questo punto la fase di prima configurazione del software dell'interfaccia è andata a buon fine, basta uscire dalle varie finestre intermedie, cliccando ok, quindi anche da quella principale (figura 13) cliccando su ok. Il programma si riduce in "system tray", cioè vicino all'orologio (figura 20). L'icona relativa a WinLirc è un piccolo bottone di solito grigio, che diviene verde alla ricezione di un comando riconosciuto da remoto, mentre è rosso se tale comando non è riconoscibile, mentre è giallo in fase di avvio del programma, quindi di inizializzazione dell'hardware.
Abbiamo completato la prima fase di configurazione software del nostro progetto, in sostanza disponiamo di un software sempre in attesa di comandi sotto forma di eventi provenienti dall'interfaccia infrarossi.
WinLIRC non fa altro che riconoscere l'evento ed indicare ad un potenziale programma in attesa sul porto 8765 (appartenente alla lista contenuta nel precedente paragrafo) il comando inviato da remoto.

Struttura di un file di configurazione .cf

Il file di configurazione "mio.cf", prodotto dal programma, contiene diverse informazioni circa il tipo di telecomando configurato, tale file la cui struttura è indicata in questa finestra di approfondimento può contenere più telecomandi così da riconoscere tutti quelli disponibili in casa. Per avere un file che dia la possibilità di riconoscere i codici di più telecomandi, basta operare più volte l'operazione di configurazione, avendo cura di salvare il contenuto del file di configurazione di volta in volta ottenuto in file separati, quindi incollare i diversi risultati in uno stesso file uno dopo l'altro.

```
# this config file was automatically
# generated using WinLIRC 0.6.5
# (LIRC 0.6.1pre3) on Sun Aug 07 21:58:06 2005
#
# contributed by
#
# brand:                Mitsubishi RM16-36605
# model:
# supported devices:

begin remote
  name  Mitsubishi RM16-36605
  bits              8
  flags SPACE_ENC
  eps              25
  aeps            100
  one             355   2071
  zero            355    852
  ptrail          355
  pre_data_bits     8
  pre_data        0xEA
  gap           24910
  toggle_bit        0

      begin codes
      Power            0x0000000000000040
      1                0x0000000000000000
      2                0x0000000000000010
      3                0x0000000000000008
      4                0x0000000000000018
      5                0x0000000000000004
      6                0x0000000000000014
      7                0x000000000000000C
      8                0x000000000000001C
      9                0x0000000000000080
      0                0x0000000000000090
      Menu              0x000000000000004A
      Cancel           0x00000000000000D0
```

10-11 diciembre 2005

Programa Marco de Formación Permanente del Profesorado

Programa de Formación Permanente del Profesorado

Programa de Formación Permanente del Profesorado

Programa de Formación Permanente del Profesorado

Lugar

Aula

8:00-15:00

15:00-19:30

```
        Select          0x00000000000000CC
        Trans           0x00000000000000AC
        *               0x0000000000000088
        Eject           0x000000000000006C
        su              0x0000000000000050
        giù             0x0000000000000048
        Rec             0x0000000000000030
        DataOnScreen    0x00000000000000E4
        Stop            0x0000000000000028
        Pause           0x0000000000000038
        Next            0x00000000000000DC
        Rew             0x0000000000000034
        Play            0x0000000000000020
        FF              0x0000000000000024
        CounterZeroStop  0x00000000000000B0
            <||          0x000000000000003C
            ||>          0x000000000000002C
            -           0x0000000000000068
            +           0x0000000000000070
            <<          0x00000000000000B2
            >>          0x00000000000000A2
        end codes
end remote
```

Si notano i codici espressi in notazione esadecimale relativi ai diversi tasti disponibili sul telecomando, mentre nella prima parte ci sono dati relativi al tipo di protocollo implementato per la comunicazione IR tra il ricevitore ed il telecomando.

Installazione e configurazione di FlexRemote 2.1.8

FlexRemote prodotto dalla ChetSoft, è un software freeware di pochi kilobyte (412Kb), che ci permette di utilizzare una qualsiasi periferica compatibile con WinLIRC, per comandare un'applicazione che accetti comandi attraverso combinazioni di tasti sotto Windows. Oltre a ricevitori autocostruiti ed in generale tutti quelli previsti da WinLIRC, il solo FlexRemote ci permette di utilizzare anche il ricevitore IR incluso in schede TV dell'AverMedia:

- TV/FM tuner - AVer TV Studio Model 203.
- TV/FM tuner - AVer TV Studio Model 305/307.
- Behold TV 401-409 [FM].

Come prima operazione dobbiamo scaricare da *www.flexremote.com* l'archivio zip "FlexRemote.zip" contenente il software, quindi il file winlirc.dll, che serve a far comunicare WinLIRC con FlexRemote. Dopo aver decompresso l'archivio zip avremo un file eseguibile di installazione FlexRemoteSetup.exe, che andremo a lanciare. Alla fine dell'installazione abbiamo tutto nella directory C:\Programmi\FlexRemote, bisogna raggiungere tale cartella e all'interno di questa, la cartella Drivers, dove dobbiamo inserire il file scaricato winlirc.dll. Ora siamo pronti per poter configurare il software per comandare le diverse applicazioni di Windows.

Figura 19 *FlexRemote 2.1.8*

Figura 20 *Icone in system tray di FlexRemote e WinLIRC (da sinistra a destra)*

Figura 21 *Voce "Localization"*

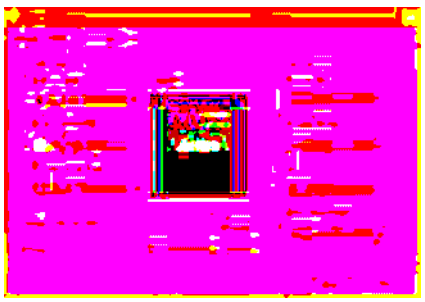


Figura 22 *Voce "Additional"*

Avviato il programma attraverso il link posto nel menù programmi (gruppo FlexRemote) o sul desktop (qualora sia stato scelto durante l'installazione), si avvierà il programma in system tray (minimizzato vicino all'orologio)(figura 20), cliccando sull'icona si apre la finestra di configurazione. In questa finestra si notano in alto a sinistra una serie di cartelle sovrapposte, l'ultima "About" è quella riportata in figura 19, quelle che ci interessano sono "General", dove troviamo le impostazioni di configurazione delle diverse applicazioni e "Additional", dove configureremo le impostazioni circa il menù da visualizzare a video. Infine c'è "Localization" che, invece, serve ad impostare la lingua (per ora sono disponibili solo inglese e russo) e il tipo di ricevitore utilizzato, naturalmente bisogna indicare WinLIRC come indicato in figura 21.
Passiamo a configurare le voci in "Additional" (figura 22). Il funzionamento del programma si basa su di un menù visualizzabile a video a seguito della pressione di un ben determinato tasto, programmato in questa schermata. Tale menù contiene tutti i programmi che abbiamo deciso poter richiamare automaticamente da telecomando, tali applicativi vengono ad essere definiti nella schermata "General".
In questo quadro abbiamo da configurare i tasti attraverso cui richiamare il menù, e navigare o selezionare una voce dello stesso.
I diversi tasti disponibili nei menù a tendina riportati sono quelli impostati in fase di addestramento di WinLIRC, e devono essere indicati dall'utente nel sottoquadro "Menu controls", poi abbiamo da indicare i colori del menù, in questo caso si è scelto sfondo nero, voce selezionata rossa, voce non selezionata bianca, infine in basso ci sono altre due opzioni, la prima circa il numero di voci nella singola schermata del menù (al massimo è possibile indicare 4) e RCSpeed che rappresenta un parametro relativo alla frequenza di campionamento del telecomando.

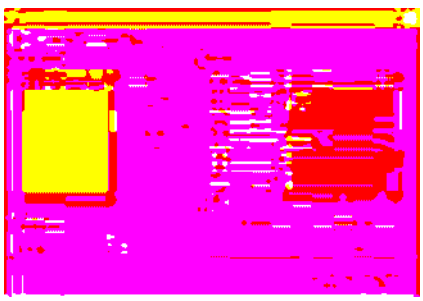


Figura 23 Voce "General"

Concludiamo con la parte più interessante del software, la composizione della lista dei programmi richiamabili da remoto e i possibili comandi assegnati ai diversi tasti.
Schematizziamo la figura 23.
Abbiamo sulla parte sinistra l'elenco dei programmi, che possiamo aggiungere, cambiare ed eliminare attraverso i tre tasti, mentre sulla destra abbiamo per lo specifico programma selezionato nella lista a sinistra, la corrispondenza tra i tasti del telecomando (quelli acquisiti da WinLIRC) e le combinazioni di tasti del programma, così da svolgere operazioni complesse come la chiusura del programma, ma anche l'avvio dell'esecuzione di un file, eccetera. Per poter programmare un tasto nell'elenco a destra, basta selezionarlo e digitare sulla tastiera la combinazione di tasti corrispondenti al comando da eseguire.
Facciamo un esempio pratico. Consideriamo uno dei più famosi media player, WinAmp della Nullsoft Inc.. Per poter avviare l'esecuzione di un file presente in playlist c'è il comando Play dell'interfaccia del programma, a tale comando corrisponde il tasto "X", a cui vogliamo collegare il tasto del telecomando Play, quindi selezioniamo il tasto del telecomando nella lista (figura 23), digitiamo X sulla tastiera, salviamo la modifica alla corrispondenza tra tasti del telecomando e combinazioni di tasti con il tasto "Save" dell'interfaccia di FlexRemote. Quanto descritto è possibile farlo con tutti i possibili comandi del programma, in figura 23 oltre a singoli caratteri ci sono anche combinazioni, per esempio al tasto "DataOnScreen" si fa corrispondere la combinazione CTRL+J (sempre relativa a WinAmp selezionato a sinistra). Con il procedimento descritto è possibile automatizzare tutta la gestione degli applicativi di Windows.

SITI DI RIFERIMENTO

- *www.lirc.org* – Sito del progetto dove è possibile trovare la documentazione e le informazioni riguardanti hardware, software del progetto;
- *usbirboy.sourceforge.net* - Sito dove è possibile trovare informazioni circa il ricevitore per porta USB mostrato in figura 4;
- *lirc.sourceforge.net/remotes* - I telecomandi supportati dal progetto;

- *winlirc.sourceforge.net* - Sito dell'equivalente per Windows di LIRC.
- *www.flexremote.com* - Sito dove troviamo FlexRemote.

CONCLUSIONI

Abbiamo visto i passaggi salienti di una delle più semplici realizzazioni di interfaccia IR-PC, che è possibile utilizzare con il progetto LIRC, poi abbiamo curato gli aspetti di installazione, configurazione e addestramento delle componenti software a corredo del ricevitore autocostruito, si spera aver fatto cosa gradita a quanti avevano chiesto più volte spiegazioni su questo tipo di applicazioni, in ogni caso per qualunque chiarimento supplementare potete scrivere a *mailbox@farelettronica.com*, i diversi dubbi saranno pubblicati anche a beneficio degli altri lettori.

Spegnere Windows XP da Telecomando

È possibile impostare FlexRemote per spegnere il computer automaticamente con la pressione di un tasto, descriveremo in questo paragrafo passo passo cosa impostare per chiudere windows automaticamente da remoto. Per maggiore completezza abbiamo bisogno di aprire una piccola parentesi su Windows XP.
Windows XP dispone al suo interno di un programma chiamato shutdown.exe (collocato nella cartella "system32") che permette di riavviare o spegnere rapidamente il personal computer. Per farlo partire, bisogna andare su Start - Esegui e digitare semplicemente: shutdown -s
A questo punto il sistema verrà automaticamente chiuso.
Ci sono però altre particolari opzioni che si possono utilizzare con il comando "shutdown". La sintassi del comando è:

```
shutdown [-l | -s | -r | -a] [-f] [-t xx] [-c "commento"] [-d up:xx:yy]
```

-i	Visualizza l'interfaccia GUI, deve essere la prima opzione
-l	Disconnette utente (non può essere utilizzato con l'opzione -m)
-s	Arresto del sistema
-r	Arresta e riavvia il sistema
-a	Interrompe l'arresto del sistema
-t	xx Imposta timeout per chiusura a xx secondi
-c	"commento" Commento per l'arresto (massimo 127 caratteri)
-f	Forza la chiusura delle applicazioni in esecuzione
-d	[u][p]:xx:yy Il codice motivo per l'arresto
u	è il codice utente
p	è il codice di un arresto pianificato
xx	è il codice motivo maggiore (numero intero positivo inferiore a 256)
yy	è il codice motivo minore (numero intero positivo inferiore a 65536)

Per poter richiamare il comando "shutdown –s" basta creare un file batch con la stringa: shutdown –s. Quindi creare con Notepad un file testo contenente tale stringa e salvare il file come "Spegni.bat", passiamo ora a FlexRemote. Aggiungiamo una nuova applicazione (vedi figura 26). Basta indicare il percorso del file batch precedentemente preparato, nel nostro caso avendo posto il file sul desktop: C:\Documents and Settings\Enrico Raffone\Desktop\spegni.bat.
Apparirà quindi una nuova voce nel menù, indicata con "Spegni WinXP". Siamo pronti per poter utilizzare questa nuova funzionalità del PC, cioè lo spegnimento automatico da remoto.

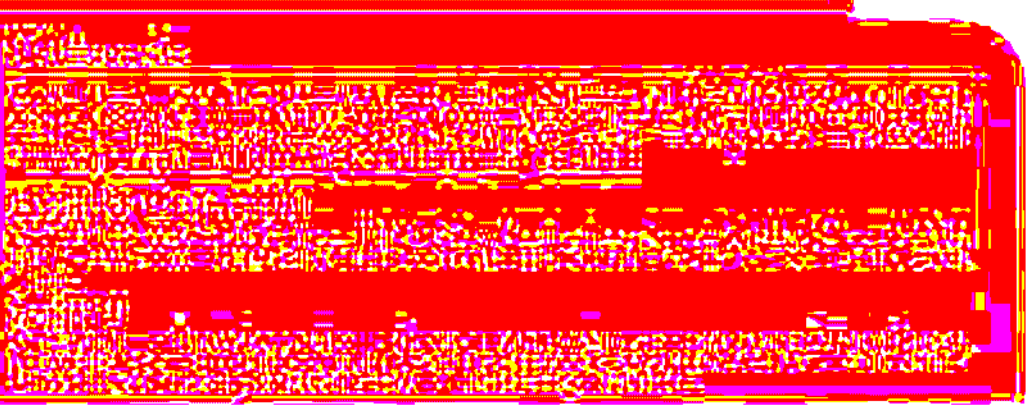
Figura 24
Il file batch "spegni.bat"

Figura 25
Inserimento nuovo programma

Figura 26
Programma spegnimento inserito

Localizzatore GPS/GSM portatile

Localizzatore GPS/GSM per ambiente



Settima parte
n° 244 - Ottobre 2005
L'operazionale in corrente alternata

Ottava parte
n° 245 - Novembre 2005
L'amplificatore differenziale

Nona parte
n° 246 - Dicembre 2005
L'operazionale a reazione di corrente

L'amplificatore operazionale dalla A alla Z

L'amplificatore operazionale utilizzato come stadio differenziale rivela già una prima, semplice applicazione del componete quale operatore matematico.

8.1 GENERALITÀ. AMPLIFICATORE DIFFERENZIALE CON UN OP-AMP

L'amplificatore operazionale, amplificando comunque la differenza fra le tensioni presenti ai due terminali di ingresso, trova una sua caratteristica applicazione nella realizzazione degli amplificatori differenziali a uno o più stadi, sia in continua che in alternata.

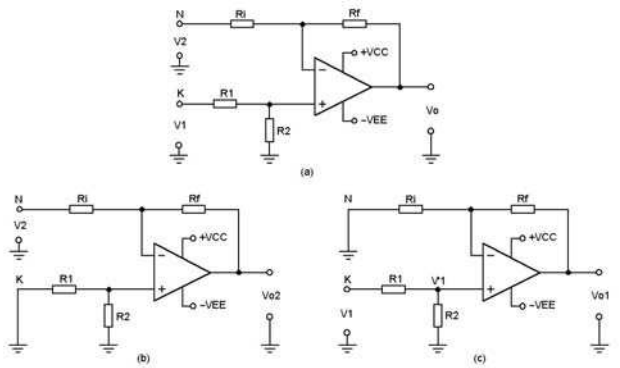


Figura 8.1 *La funzione di trasferimento dell'amplificatore differenziale con operazionale (a) si può ricavare considerando separatamente i circuiti (b) e (c) e quindi sovrapponendo i risultati*

Si consideri, per esempio, il circuito della figura 8.1 *a* costituito da un unico operazionale e quattro resistenze. Il metodo più rapido per ricavare la funzione di trasferimento fa ricorso al principio di sovrapposizione degli effetti. Si considerano quindi, separatamente, la risposta V_{o2} al segnale di ingresso V_2 mentre l'ingresso che fa capo all'input non invertente dell'op-amp (punto *K*) è a massa, e quindi la risposta V_{o1} al segnale V_1 mentre è a massa l'ingresso che fa capo all'input invertente (punto *N*).

Questa soluzione implica pertanto, lo studio dei due circuiti riportati nelle figure 8.1 *b* e *c*. Con riferimento quindi alla figura 8.1 *b* si vede subito che la configurazione è quella dell'op-amp invertente e pertanto per la tensione di uscita V_{o2} si ha:

$$V_{o2} = -V_2 \cdot \frac{R_f}{R_i} \qquad [8.1]$$

Lo stadio amplificatore della figura 8.1 *c* rappresenta un op-amp in configurazione non invertente. Pertanto la tensione V'_1 a valle del partitore R_1-R_2 , ossia la tensione sul terminale (+) dell'operazionale, ha espressione:

$$V'_1 = V_1 \cdot \frac{R_2}{R_1 + R_2} \qquad [8.2]$$

Per la tensione V_{o1} di uscita corrispondente alla tensione di ingresso V_1, tenendo conto della [8.2], si avrà quindi:

$$V_{o1} = V'_1 \cdot \left[1 + \frac{R_f}{R_i}\right] = \frac{(V_1 \cdot R_2)\cdot(R_i + R_f)}{R_i \cdot (R_1 + R_2)} \qquad [8.3]$$

Poiché, sovrapponendo gli effetti, per la tensione risultante V_o vale l'espressione:

$$V_o = V_{o1} + V_{o2}$$

sommando la [8.1] e la [8.3] si ricava:

$$V_o = V_1 \cdot \frac{(V_1 \cdot R_2)\cdot(R_i + R_f)}{R_i \cdot (R_1 + R_2)} - V_2 \cdot \frac{R_f}{R_i}$$

Ponendo in questa $R_1 = R_i$ e $R_f = R_2$ si ha:

$$V_o = -(V_2 - V_1) \cdot \frac{R_f}{R_i} \qquad [8.4]$$

L'espressione [8.4] indica che la tensione di uscita è funzione diretta della differenza fra le tensioni applicate ai due ingressi dell'amplificatore operazionale e che, con riferimento al guadagno, l'op-amp nella configurazione differenziale si comporta come amplificatore invertente. Dividendo il

L'amplificatore differenziale

di Nico Grilloni
(n.grilloni@farelettronica.com)

primo e secondo membro della [8.4] per ($V_2 - V_1$) si ha infatti:

$$A_{CL} = V_o / (V_2 - V_1) = -(R_f / R_i) \qquad [8.5]$$

In simulazione al computer, l'amplificatore di cui alla figura 8.1 *a* – posto $R_1 = R_i = 1$ kΩ e $R_f = R_2 = 100$ kΩ (quindi $A_{CL} = 100$) – si è pilotato con segnali sinusoidali fra loro in fase e di ampiezza $V_1 = 1$ V da picco a picco e $V_2 = 1{,}1$ V. Essendo ($V_2 - V_1$) = 0,1 V e $A_{CL} = 100$ si è ottenuta in uscita, coerentemente con la [8.5], una tensione di 10 V da picco a picco come mostra la figura 8.2. Qui si nota lo sfasamento di 180° fra l'uscita e l'ingresso, sfasamento determinato dalla condizione $V_2 > V_1$. Se, viceversa, si fosse imposto $V_1 > V_2$, la tensione di uscita sarebbe stata in fase con l'ingresso. Il secondo membro dall'espressione [8.4], per $V_1 > V_2$ diviene infatti positivo.

8.2 AMPLIFICATORE DIFFERENZIALE CON TRE OPERAZIONALI

Una delle più importanti e utilizzate caratteristiche dell'amplificatore differenziale è la capacità di approssimare a zero segnali spuri, ossia segnali ad andamento del tutto casuale, determinati da fenomeni induttivi che si manifestano in prossimità dell'amplificatore.
Poiché questi segnali si presentano a entrambi gli ingressi – segnali di modo comune – con le stesse caratteristiche elettriche e/o elettromagnetiche, la tensione a cui danno origine in uscita, a condizione che le resistenze dei due ingressi siano il più possibile eguali fra loro, è, se non nulla, di ampiezza assai prossima a zero.

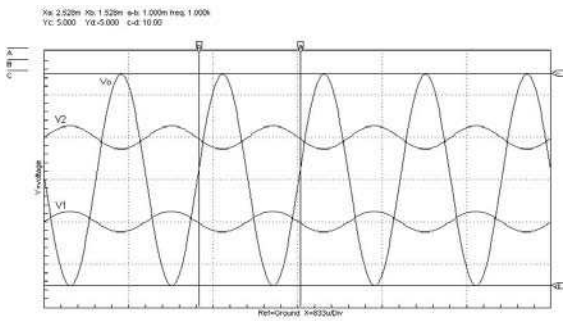


Figura 8.2 *La tensione V_o di uscita è eguale a A_{CL} ($V_2 - V_1$), dove V_1 e V_2 sono le tensioni applicate ai due ingressi dell'amplificatore differenziale della figura 8.1*

L'amplificatore differenziale di cui alla figura 8.1 *a*, guardando dai terminali a cui si connettono le sorgenti di segnale V_1 e V_2, presenta invece sull'ingresso invertente una resistenza $R_i = 1$ kΩ dal momento che l'input (–) dell'op-amp è a massa virtuale, mentre presenta sull'ingresso non invertente una resistenza pari a ($R_1 + R_2$) essendo già, anche senza considerare l'effetto della reazione, estremamente alta la resistenza intrinseca d'ingresso dell'operazionale. Il generatore di segnale V_1 vede quindi una resistenza pari a 101 kΩ. Le resistenze di ingresso sono quindi sbilanciate e ciò comporta che l'ingresso a resistenza più elevata sia meno immune dell'ingresso a resistenza più bassa da interferenze di varia natura. L'effetto più diretto è una caduta del valore del *CMRR*.
Per ovviare a ciò si possono attribuire alle quattro resistenze valori tali che sia ($R_1 + R_2$) = R_i e $R_1/R_2 = R_i/R_f$.
Migliore è però la soluzione illustrata nella figura 8.3 dove, a monte del circuito amplificatore della figura 8.1 si sono posti due operazionali come inseguitori di tensione. Utilizzando, per esempio, l'op-amp LF353 che, in unico involucro, ha due operazionali, si ha una resistenza nominale di 10^{12} Ω su ciascun ingresso.
È evidente che la cura che si porrà nella realizzazione di un amplificatore differenziale sarà funzione della sua destinazione d'uso. Per esempio, lo stadio differenziale della figura 8.3, per il quale valgono le stesse formule viste per il differenziale della figura 8.1, è già un buon amplificatore differenziale per strumentazione. Questo può essere ulteriormente elaborato secondo la configurazione circuitale esposta nella figura 8.4.
La differenza fra questa e la precedente risiede nella diversa composizione dei due stadi di ingres-

so. Questi sono comunque inseguitori di tensione ma, contrariamente ai voltage follower classici, consentono di modificare il guadagno ponendo al posto della resistenza R_4 una resistenza fissa con un trimmer in serie.
Per ricavare l'espressione del guadagno si può sempre ricorrere al principio di sovrapposizione degli effetti. Indicando allora con V_M la tensione presente sull'uscita dell'operazionale A_1 e dovuta al segnale di ingresso V_1, si ha:

$$V_M = V_1 \cdot \left[1 + \frac{R_5}{R_4+R_3}\right] - V_N \cdot \frac{R_3}{R_3+R_4} = V_1 \cdot \frac{R_5+R_3+R_4}{R_3+R_4} - V_N \cdot \frac{R_3}{R_3+R_4}$$

Per la tensione V_N presente sull'uscita dell'operazionale A_2 e dovuta al segnale di ingresso V_2, si ha:

$$V_N = V_2 \cdot \left[1 + \frac{R_3}{R_4+R_5}\right] - \frac{V_M \cdot R_3}{R_3+R_4} = V_2 \cdot \frac{R_5+R_3+R_4}{R_4+R_5} - \frac{V_M \cdot R_5}{R_4+R_5}$$

Ponendo nelle ultime due espressioni $R_5 = R_3$, e sottraendo la seconda dalla prima, si ottiene:

$$V_M - V_N = \frac{R_4+R_5}{R_4} \cdot \left[V_1 \cdot \frac{R_4+2R_5}{R_4+R_5} - V_2 \cdot \frac{R_4+2R_5}{R_4+R_5}\right]$$

da cui:

$$A'_{CL} = \frac{V_M - V_N}{V_1 - V_2} = 1 + \frac{2R_5}{R_4}$$

Questo è il guadagno differenziale degli stadi A_1 e A_2. Tenendo conto dell'ulteriore guadagno dello stadio di uscita A_3 che, per la [8.5], è pari al rapporto (R_f/R_i) e ricordando che il guadagno complessivo A_{CL} è dato dal prodotto dei guadagni, in definitiva si ha:

$$A_{CL} = \left[1 + \frac{2R_5}{R_4}\right] \cdot \frac{R_f}{R_i} \qquad [8.6]$$

Da questa espressione si vede come ponendo per R_4 una resistenza in serie con un trimmer si possa variare il guadagno del differenziale.
Per esempio, per i valori attribuiti alle resistenze nel circuito della figura 8.4, al posto della R_4 si può mettere una resistenza da 1 kΩ con in serie un trimmer da 22 kΩ. Così facendo il guadagno A_{CL} potrà variare da un minimo di 269 a un massimo di 1220). In simulazione, con i valori delle resistenze riportati nella figura 8.4, per A_{CL} si è ricavato:

$$A_{CL} = \left[1 + \frac{44000}{12000}\right] \cdot \frac{100}{1} = 466{,}66$$

Avendo poi pilotato gli ingressi con tensioni sinusoidali, ed esattamente con V_1 = 1020 mV da picco a picco e V_2 = 1000 mV da picco a picco, si ha ($V_1 - V_2$) = 20 mV. Moltiplicando questo valore per A_{CL} = 466,66 si ha l'ampiezza da picco a picco del segnale V_o di uscita. È quindi:

$$V_o = 20 \times 10^{-3} \times 466{,}66 = 9{,}33 \text{ V}$$

La figura 8.5 mostra a tal proposito i segnali V_1 e V_2 di ingresso e il segnale V_o di uscita. Questo, come indica la posizione reciproca dei marker orizzontali, ha ampiezza di 9,33 V. Si legge infatti, ($c - d$) = 9,333 V.

8.3 L'AMPLIFICATORE DIFFERENZIALE IN ALTERNATA

Per un uso specifico in alternata dell'amplificatore operazionale è però opportuno porre sugli ingressi un filtro passa-alto passivo, anche un semplice *R-C*, con la resistenza *R* fra gli ingressi non invertenti e massa di valore non inferiore a 10 MΩ. Il valore della capacità *C* si calcolerà con le espressioni già illustrate nell'articolo precedente in funzione della frequenza di taglio inferiore e della resistenza *R*. Così modificato, ossia ponendo in ingresso i filtri passivi riportati nella figura 8.6, l'amplificatore differenziale della figura 8.4 può costituire, per esempio, l'elemento base di un elettrocardiografo ponendo *R* = 10 MΩ e C = 0,33 μF. Per questa utilizzazione è poi necessario portare il guadagno complessivo a circa 1000 ponendo $R_i = R_1$ = 10 kΩ, $R_5 = R_3$ = 51 kΩ e R_4 = 1 kΩ. Con questi valori si ha un guadagno A_{CL} pari a 1030. Un valore così elevato è necessario dal momento

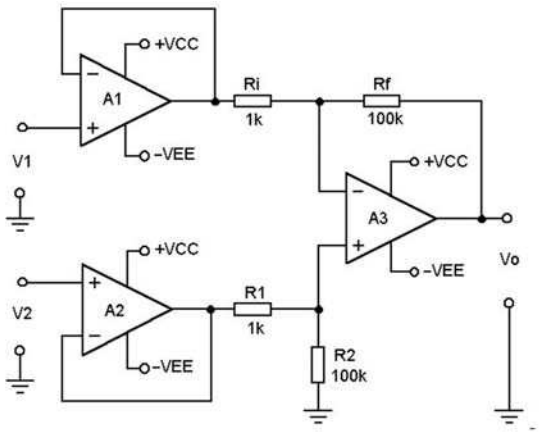


Figura 8.3 *Amplificatore differenziale con tre operazionali. Poiché A_1 e A_2 sono dei voltage follower, le resistenze di ingresso sono elevate e pressoché eguali*

che il segnale disponibile all'elettrodo posto sull'epidermide è di circa 1 mV mentre il segnale utile ai sistemi di registrazione dei tracciati elettrocardiografici dev'essere mediamente non inferiore a 1 V. L'esigenza di porre in ingresso i condensatori deriva anche dalla necessità di "tagliare" la tensione continua di natura elettrolitica che si crea nel contatto fra gli elettrodi [*GD* (gamba destra), *BD* (braccio destro), *BS* (braccio sinistro)] e l'epidermide, tensione che può anche essere di 2 V, ossia di circa 2000 volte maggiore del segnale utile. I condensatori, nel caso dell'elettrocardiografo, sono dimensionati per una frequenza bassissima. Con i valori della figura 8.6 si ha infatti:

$$f_L = 0{,}159 / (10 \times 10^6 \times 0{,}33 \times 10^{-6}) = 0{,}048 \text{ Hz}$$

Gli elettrocardiografi commerciali coprono generalmente un campo di frequenze di funzionamento compreso fra 0,05 Hz e 100 Hz o fra 0,05 Hz e 40 Hz. In questi strumenti, anche con destinazione d'uso diversa, la resistenza R_2 del circuito della figura 8.4 viene sostituita da una resistenza da 100 kΩ in serie a un trimmer multigiri da 50 kΩ, come mostra la figura 8.7, al fine di aumentare quanto più è possibile il *CMRR*.

La procedura per esaltare il *CMRR* è semplice: dapprima si pongono a massa gli ingressi; si connette poi in uscita, ossia fra il terminale di uscita e massa, un voltmetro in continua con 1,5 V di fondo scala; si agisce quindi sul trimmer fino ad avere in uscita 0 V. Via via che la tensione di uscita si abbassa è opportuno, al fine di minimizzare l'errore, portare il voltmetro su valori di fondo scala via via minori. Nel caso dell'elettrocardiografo ci sono ulteriori fasi: portata a zero l'uscita, sempre con riferimento alla figura 8.7, si sconnette l'ingresso *GD* da massa e si connette un generatore di segnale fra l'ingresso *GD* e gli ingressi *BS* e *BD* ancora connessi a massa. Si fa erogare al generatore un segnale sinusoidale con frequenza variabile fra 10 e 50 Hz, una tensione di 1 o 2 V di picco. A questo punto, connesso in uscita un voltmetro in alternata (o l'oscilloscopio), si agirà sul trimmer R_p fino ad avere un segnale minimo.

PER IL LETTORE

I diagrammi qui riportati sono stati ricavati con il software di simulazione CircuitMaker della Microcode Engineering Inc. – Utah – USA, che utilizza SPICE (*Simulation Program with Integrated Circuit Emphasis*) realizzato dalla Microcode Engineering Inc. (USA).
Sito Internet: www.microcode.com.

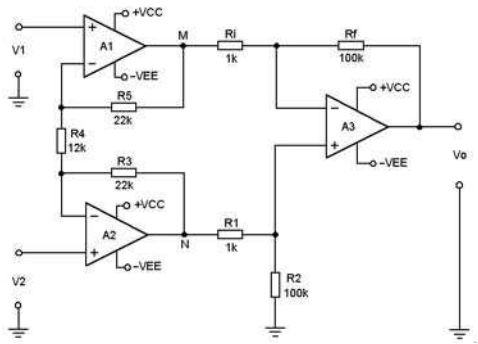


Figura 8.4 *Amplificatore differenziale per strumentazione con tre operazionali che, valido anche in alternata, si può rendere a guadagno variabile agendo sul valore della R_4*

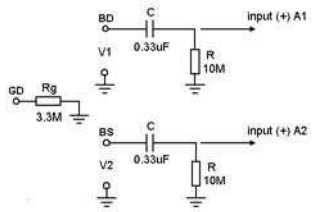


Figura 8.6 *Utilizzando l'amplificatore della figura 8.4 in alternata è necessario porre sui due ingressi i filtri passivi RC*

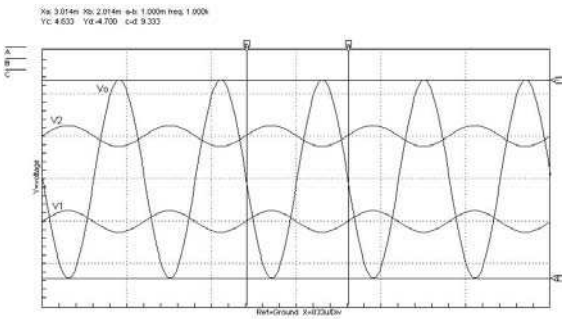


Figura 8.5 *Con riferimento all'amplificatore della figura 8.4, la tensione di uscita Vo ha un'ampiezza pari a circa 467 volte la differenza delle tensioni V_1 e V_2 applicate agli ingressi.*

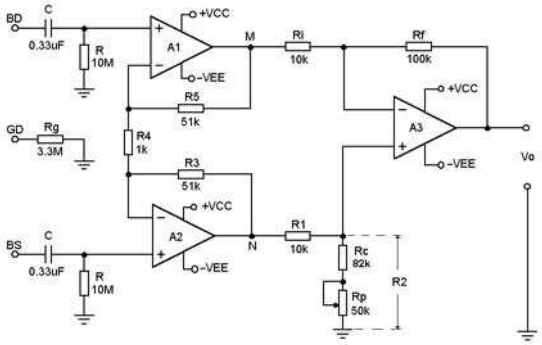


Figura 8.7 *Amplificatore differenziale in funzione di elettrocardiografo*

Prima parte
n° 244 - Ottobre 2005
Evoluzione dei componenti programmabili

Seconda parte
n° 245 - Novembre 2005
Utilizzare le CPLD XC9500 e l'ambiente di sviluppo Xilinx

Terza parte
n° 246 - Dicembre 2005
Sistema hardware di sviluppo

CPLD by Example

Nella prima puntata abbiamo svolto una panoramica sull'evoluzione dei componenti programmabili e in particolare sulle CPLD, soffermandoci sulle serie Xilinx. In questa seconda parte trattiamo in maniera approfondita la struttura dei dispositivi XC9500 e l'ambiente di sviluppo ISE WebPack. Gli argomenti sono presentati attraverso alcuni esempi pratici.

MODULO 1

CPLD XC9500

Analizziamo ora l'architettura interna delle CPLD della serie XC9500. Le considerazioni che faremo sono di carattere generale e sono quindi valide per tutte le CPLD di questa serie.
Occorre tenere presente che, da un componente all'altro, ciò che cambia è, in sostanza, l'ammontare dei gate a disposizione, il numero di macrocelle e dei pin di I/O.
I pin si suddividono in due categorie: pin di I/O ordinari e pin di controllo.
Questi ultimi comprendono il reset, le abilitazioni dei buffer tri-state di uscita, gli ingressi dei clock globali e i segnali per la programmazione in-system.

Struttura interna della CPLD

Diamo un'occhiata più da vicino all'interno del dispositivo XC9500.
La figura 1 ci presenta una vista d'assieme del top-level. Consideriamo quindi i blocchi interni e il loro significato.

Blocchi di I/O

In azzurro sono indicati i blocchi di ingresso/uscita (I/O Blocks). Nel dettaglio, ciascun blocco di I/O (figura 2) è composto da un buffer di ingresso, un buffer di uscita, un multiplexer per dirigere l'uscita e alcuni controlli.

Matrice di interconnessione

La matrice di interconnessione è evidenziata in giallo nella figura 3.
Essa controlla i segnali di ingresso ai blocchi funzionali, denominati *Function Block* in figura.
Tutti i segnali provenienti dalle porte di ingresso/uscita e quelli dei Function Block sono collegati per mezzo della matrice, che è di tipo proprietario *FastCONNECT™*.

Function Blocks

Il blocco di funzione è composto di un piano AND programmabile, di un allocatore di termini di prodotto e di una serie di macrocelle, come possiamo vedere dalla figura 4.

Programmabilità in-system

La programmabilità in-system è uno dei grandi vantaggi di questi dispositivi. La circuiteria interna al chip per l'interfaccia di programmazione è evidenziata in figura 5.
I dispositivi della serie XC9500 vengono programmati per mezzo del protocollo standard JTAG. Durante la programmazione i pin di I/O ordinari sono mantenuti a livello High.

IMPIEGHI

Che cosa si può fare con una CPLD?

Possiamo realizzare delle funzioni di logica combinatoria, come ad esempio un decoder per la generazione di chip select, o una tabella di transcodifica tra codici diversi (cioè in pratica una ROM), ma anche dei registri o dei blocchi di memoria RAM, compatibilmente con il numero di flip-flop che la CPLD contie-

Utilizzare le CPLD XC9500 e l'ambiente di sviluppo Xilinx

di Agostino Rolando
a.rolando@farelettronica.com

ne al suo interno.
Bisogna sempre tenere presente che ciò che vogliamo realizzare deve rispettare la struttura interna del nostro dispositivo. Non potremo richiedere prestazioni impossibili; tuttavia, con l'evolversi dei dispositivi, è prevedibile che i limiti fisici verranno via via superati. Un aspetto significativo, a questo proposito, è rappresentato dalla possibilità di stabilire il *pin-lock* (cioè di fissare il pinout del dispositivo) a proprio piacimento.
Per quanto riguarda la logica sequenziale, si possono realizzare macchine a stati finiti, contatori, divisori, shift registers, eccetera. Inoltre, anche alcune funzioni complesse possono trovare posto entro la nostra CPLD, ad esempio controllori, uart, eccetera.

ESEMPIO 1: ACCENDERE UN LED

Schema elettrico

Per questo esempio, data la semplicità della realizzazione e trattandosi di uno schema di principio, non forniamo il circuito stampato, ma solo lo schema elettrico (figura 6).
Per realizzarlo possiamo impiegare una basetta preforata e montare la CPLD su zoccolo. Il componente scelto è la XC9536, la più "piccolina" della famiglia, che è reperibile in package PLCC a 44 pin.
All'interno contiene due *Function Blocks* di tipo 36V18, per un ammontare di 800 gates, come possiamo vedere dall'architettura in figura 7. Il periodo di propagazione dei segnali è di 5 ns e la frequenza di lavoro può raggiungere i 100 Mhz.

Per il funzionamento del circuito l'alimentazione a +5V viene fornita per mezzo del connettore J01.
La corrente massima erogabile per pin in uscita è pari a 24 mA, più che sufficiente a pilotare un led.

Significato dei pin

Si può prendere spunto da questo schema per approfondire il significato del pinout della XC9536. I pin denominati VCCINT si riferiscono all'alimentazione per la logica e per i buffers interni, mentre al pin VCCIO va data l'alimentazione

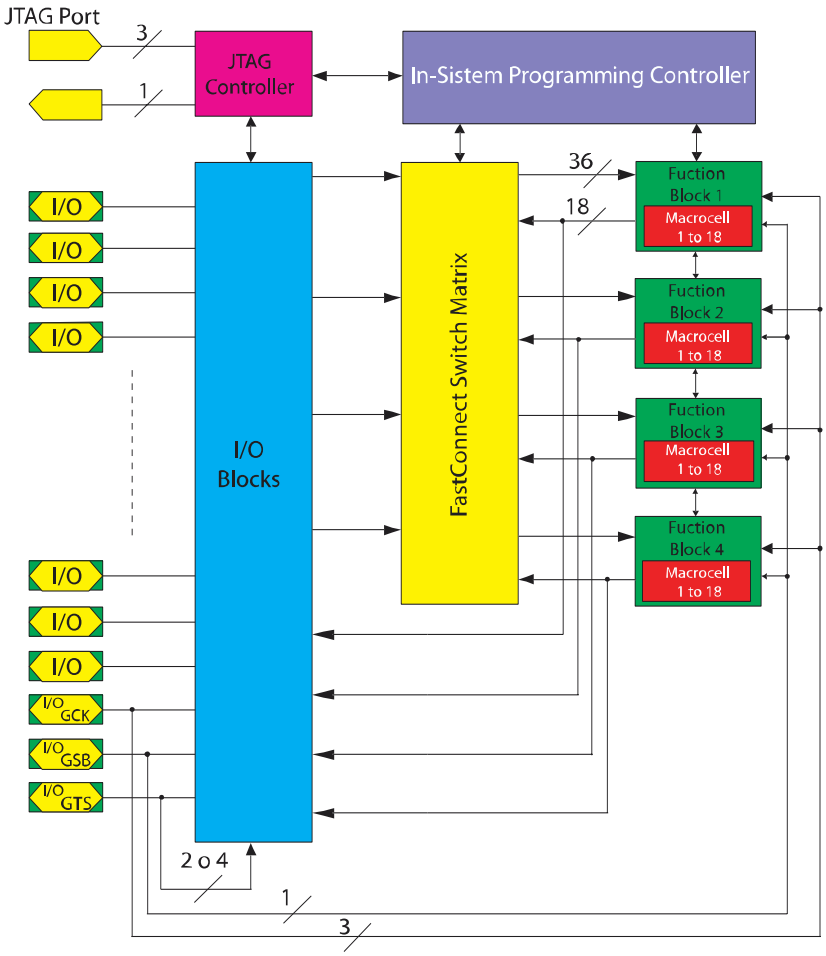


Figura 1 Diagramma a blocchi del dispositivo XC9500

per i drivers di uscita. Vi sono altri pin, denominati *Globali*, che hanno il seguente significato:
GSR è il pin di *Global Reset*. Un impulso a livello basso all'accensione su questo pin resetta il dispositivo. Nello schema il reset viene prodotto con il gruppo R01-C02. Per poter fornire un reset manuale è stato inserito il pulsante P01.
GCK1/2/3 sono i pin di ingresso dedicati per i *clock principali* (*Global Clock*). Tre sono le possibili sorgenti di clock collegabili al dispositivo anche contemporaneamente. Nel nostro esempio abbiamo inserito all'ingresso GCK1 il pulsante P02, a scopo didattico. Con questo sistema potremo visualizzare comodamente il comportamento passo-passo della nostra CPLD, in modo da testare l' esempio di codice che seguirà.
Rimangono da considerare i pin di controllo GTS1/2. (*Global Tri-State*). Essi servono per abilitare i buffers tri-state di uscita. Sono attivi bassi, per cui vi abbiamo collegato le resistenze di pull-down R02 e R03.

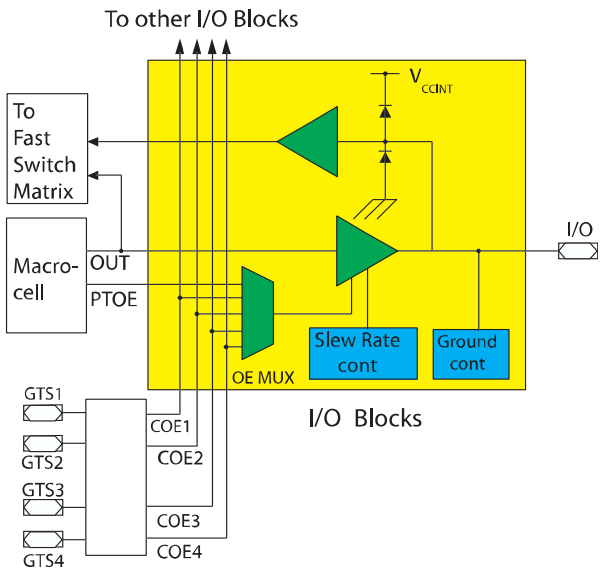


Figura 2 Blocco di I/O

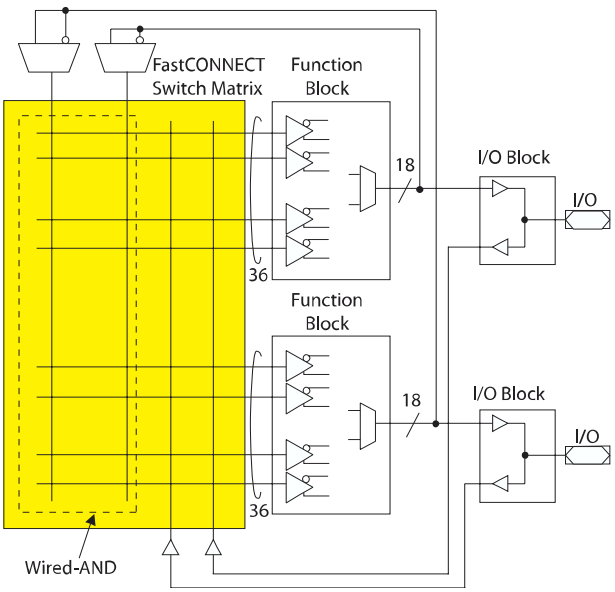


Figura 3 Matrice di interconnessione

Riguardo l'interfaccia di programmazione JTAG, nello schema elettrico abbiamo riportato i relativi pin al connettore J02. Per saperne di più sulla programmazione in-system e su come realizzare l'interfaccia possiamo fare riferimento all'articolo "Interfaccia JTAG", comparso sul numero 224 della Rivista (febbraio 2004). In ogni caso, lo schema elettrico dell'interfaccia verrà riproposto nella quarta puntata.
Possiamo notare che i pin di ingresso/uscita che la CPLD ci mette a disposizione sono parecchi, 28 per la precisione, disposti sul lato destro del dispositivo nello schema. Questi pin sono I/O di tipo general-purpose. Da notare che anche i pin di Global Clock possono essere utilizzati come I/O generico, mentre quelli del JTAG non si possono usare per questo scopo.
Nella tabella 1 sono riportati tutti i pin della XC9536, suddivisi per tipo e funzione. Sono indicati anche i Function Blocks e le Macrocelle a cui appartengono.
La figura 8 ci mostra come i pin sono distribuiti a livello del package.
In questo esempio, di tutta l'abbondanza di piedini di I/O, abbiamo volutamente utilizzato solo l' IO9, per collegarvi la resistenza R06 e il diodo led SL01. In nostro scopo è di accendere e spegnere il led in maniera bistabile, premendo il pulsante P02. Vedremo più avanti come si può fare questo in Verilog.

MODULO 2
Sistema di sviluppo Ise Webpack

Per quanto riguarda l'ambiente di sviluppo ISE di Xilinx, possiamo fare riferimento all'articolo comparso sul numero 227 di Fare Elettronica. Sul successivo numero sono descritti i passi necessari per trasferire il codice di programmazione sul chip.
Per poter operare con le CPLD Xilinx gli strumenti necessari sono costituiti da un editor per la crea-

fare elettronica

CAMPAGNA ABBONAMENTI 2005/ 2006

1 ANNO, 11 RIVISTE

A SOLI 45,00 EURO INVECE DI 60,50 EURO

con un risparmio del

25%

Abbonati subito!

Compila oggi stesso il coupon che trovi in ultima pagina o abbonati su www.farelettronica.com/abbonamento

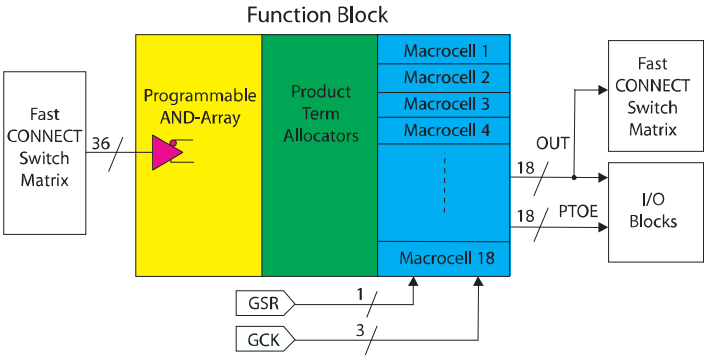


Figura 4 *Function block*

Pin number	FB	Macrocell	Pin number	FB	Macrocell
1	2	1	23	GND	
2	1	1	24	1	17
3	1	2	25	2	17
4	1	4	26	2	16
5	1	3	27	2	15
	GCK1				
6	1	5	28	2	14
	GCK2				
7	1	7	29	2	13
	GCK3				
8	1	6	30	TDO	
9	1	8	31	GND	
10	GND		32	VccIO 3.3V/5V	
11	1	9	33	2	12
12	1	10	34	2	11
13	1	11	35	2	10
14	1	12	36	2	9
15	TDI		37	2	8
16	TMS		38	2	7
17	TCK		39	2	6
				GSR	
18	1	13	40	2	5
				GTS2	
19	1	14	41	VccINT 5V	
20	1	15	42	2	3
				GTS1	
21	VccINT 5V		43	2	4
22	1	16	44	2	2

Tabella 1 *Pin diagram della XC9536*

zione del file sorgente, il tool di sintesi e l'ambiente di simulazione Modelsim.

Azioni da compiere per il download dei files di installazione

Per scaricare il pacchetto gratuito ISE WebPACK occorre prima registrarsi all'indirizzo www.xilinx.com dopodichè, una volta ottenuta la password di accesso, ci si potrà collegare alla pagina web contenente i files di installazione da scaricare.
Notiamo che nell'area di download, oltre ai files di installazione del pacchetto ISE, si può anche effettuare il download del Software di simulazione Modelsim.
Effettuati questi passi, potremo installare l'ambiente ISE (*Integrated Software Environment*) sul nostro PC. Ci verrà chiesto di inserire il codice di registrazione, che avremo previamente ricevuto via e-mail.

Utilizzo dell'ambiente

Ci soffermiamo ora sull'ambiente di sviluppo, analizzando i passi necessari per creare un nuovo progetto. L'ambiente ISE (di cui utilizziamo la versione 5.2i) riconosce tre linguaggi di programmazione HDL: Abel, Vhdl e Verilog. Prenderemo in considerazione esempi scritti in Verilog.
Attiviamo dunque il nostro *Project Navigator* (figura 9) e creiamo un nuovo progetto, inserendo tutti i dati relativi al dispositivo che abbiamo scelto, al package e al linguaggio utilizzato (figura 10):
Project name: esempio1

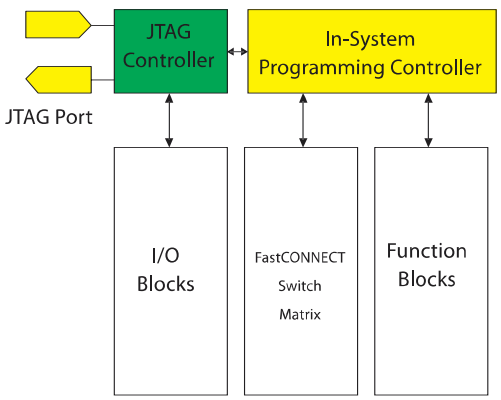


Figura 5 *Diagramma della sezione di in-system programming*

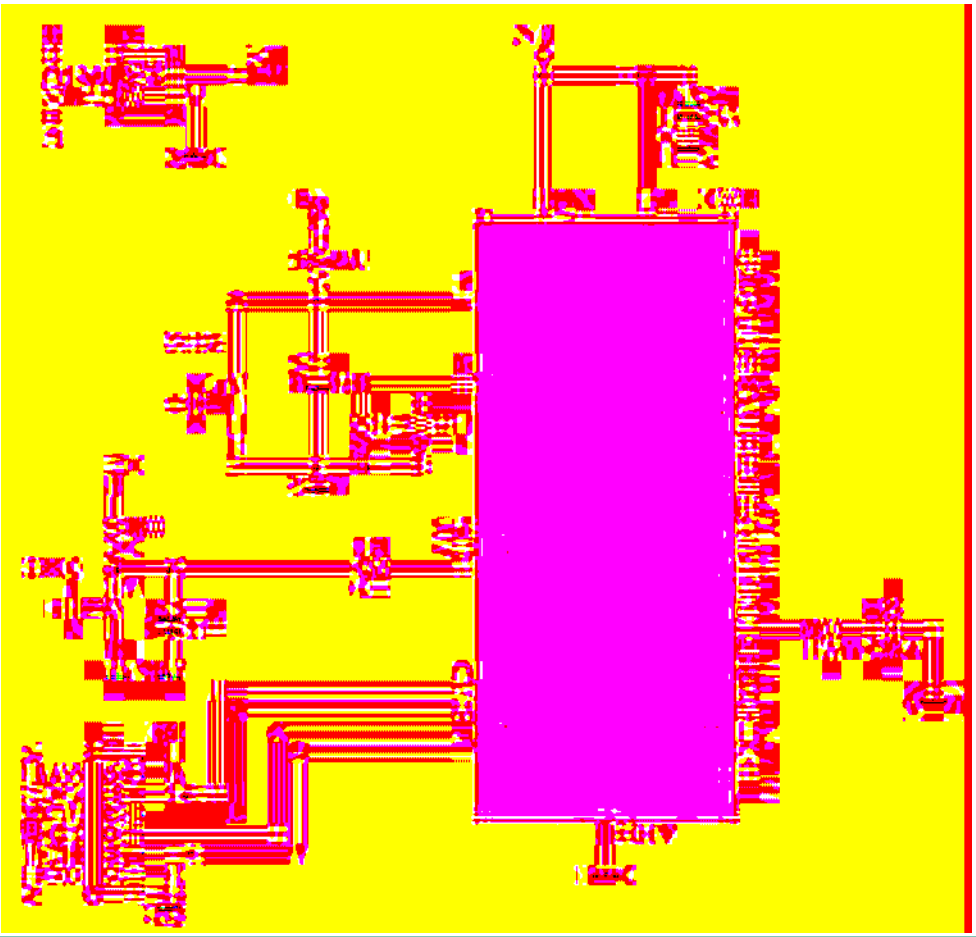


Figura 6 Schema elettrico per l'esempio 1

Elenco componenti ESEMPIO 1	
Sigla	**Valore**
R01÷R04	10 KΩ 1/4 W
R05	100 Ω 1/4 W
R06	390 Ω 1/4 W
C01, C03	0,1 µF 50 V ceramico
C02	1 µF 20 V elettrolitico
SL01	Diodo LED
IC01	CPLD XC9536-15PC44C /package 44 PIN PLCC)
J01	Connettore di alimentazione da stampato
J02	Connettore lineare a striscia 6 pin
P01,P02	Pulsante

Device Family: XC9500 CPLDs
Device: xc9536
Package: PC44
Design flow: XST Verilog

A questo punto, abbiamo definito il nostro progetto e nella schermata dell'ambiente di navigazione si presenterà il nome del file che contiene i dati di definizione del progetto, *esempio1.npl*, come da figura 11.
A questo punto, dovremo creare il file sorgente. Cliccando su *Project -> New Source* si presenterà la form di figura 12, in cui dovremo inserire il nome del file sorgente (ad esempio top) e scegliere il tipo di file, nel nostro caso "Verilog Module".
Fatto ciò, clicchiamo sui tasti *next* due volte e infine su *finish*.
Ci viene presentata una bozza di file sorgente, già pronto per l'editing con le parole chiave di "inizio" e "fine" (*cioè module* e *endmodule*).
La situazione è ora come da figura 13.

Codice sorgente

Non ci resta che scrivere il codice o importarlo dall'esterno. Innanzi tutto occorre definire i segnali di ingresso e uscita, che indichiamo *reset*, *clock* e *led*.

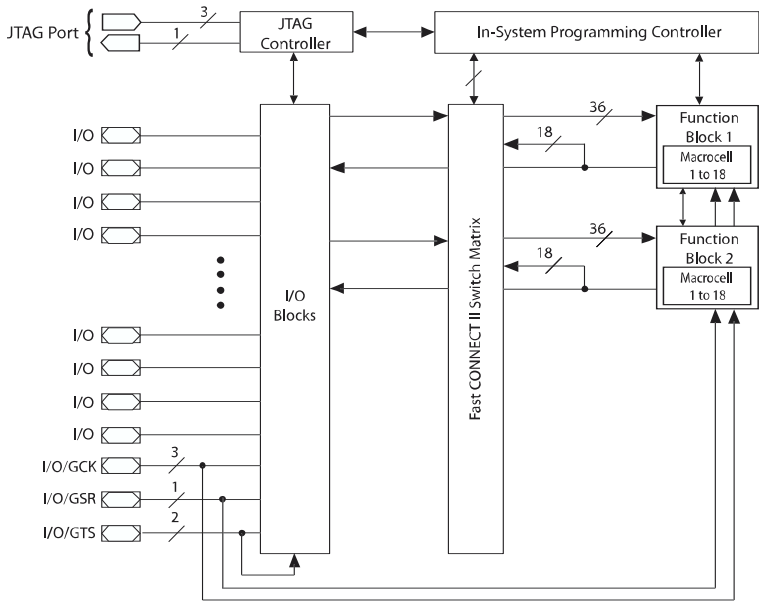


Figura 7 Architettura della XC9536

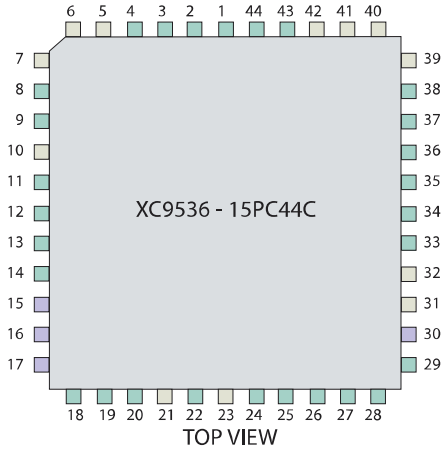


Figura 8 Pinout del dispositivo

Poi definiamo *led* come una variabile di tipo registro, che vogliamo venga tradotta in un flip flop di tipo *toggle*.
Ci basterà infine aggiungere la descrizione di un loop infinito con il costrutto *always* in cui, ad ogni colpo di clock (sul fronte di discesa), l'uscita cambierà stato, come da descrizione seguente:

```
module top(reset,clock,led);
  input reset,clock;
  output led;
  reg led;
  always @ (negedge reset or negedge clock)
  begin
    if  (!reset)
      led <= 1'b0;
    else
      led <= ~led;
  end
endmodule
```

Dunque, al reset la variabile *led* vale zero e, in corrispondenza del clock, assume di volta in volta il negato del valore precedente.

Assegnazione del pinout

Per definire il pinout del chip si procede in modo analogo alla creazione del file sorgente: cliccando su *Project -> New* Source si presenterà la stessa form di figura 12, in cui dovremo questa volta

Figura 9 Icona di ISE Project Navigator

Figura 10 Inserimento dati del nuovo progetto

Figura 11 Schermata iniziale

scegliere il tipo di file *Implementation Constraint File* (figura 14) e inserire il nome del file di constraint; per comodità possiamo chiamarlo *topucf*. Possiamo poi editare il file facendo doppio click su *Edit Constraints*.
Inseriremo le assegnazioni seguenti:

```
net CLOCK  LOC="P5";
net RESET  LOC="P39";
net LED    LOC="P13";
```

Flusso di progetto

Ora che abbiamo definito le parti fondamentali del nostro progetto, prima di procedere spendiamo due parole a proposito del *flusso di progetto ISE* per dispositivi CPLD, che vediamo descritto nello schema a blocchi di figura 15.

Descrizione comportamentale

La prima fase è quella della Descrizione comportamentale, cioè del comportamento del dispositivo ad alto livello, fatta con i costrutti del linguaggio.
La descrizione è *ideale*, nel senso che non si prevedono ritardi fisici.

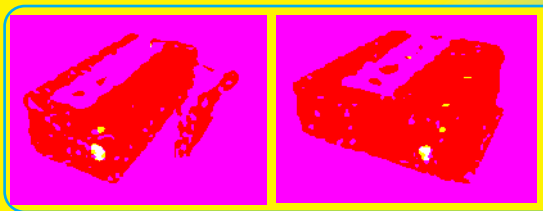
Sintesi

La successiva fase di Sintesi effettua la traduzione dalla descrizione comportamentale a quella strutturale, cioè a livello di porte logiche (netlist). La Sintesi è l'anello di congiunzione tra le due descrizioni (figura 16a) e si basa sulla presenza di librerie di simboli adeguati.
È suddivisa nelle seguenti fasi:

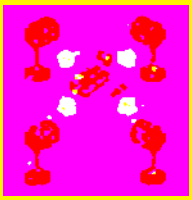
- Analisi sintattica del sorgente
- Compilazione, cioè traduzione a livello RTL (Register Transfer Level)
- Ottimizzazione, nel senso del miglioramento della logica

La Sintesi è un procedimento guidato e può fare uso di opportuni vincoli. Una trattazione completa sui vincoli (o constraints) verrà fatta in seguito.

S.V.M. ELETTRONICA
Vendita per corrispondenza Tel./Fax 0331/640569
Caratteristiche tecniche e vendita on-line **www.svmelettronica.com**



INVERTER 12VDC/220VAC SOFT-START
WHS150W-12 €. 40,00
WHS200W-12 €. 45,00
WHS300W-12 €. 52,00
WHS400W-12 €. 60,00
WHS600W-12 €. 95,00



Sistema di videosorveglianza wireless operante sulla banda dei 2,4GHz composto da 4 telecamere a colori da esterno con illuminatore IR a 12 LED, e da un ricevitore multicanale con switcher e telecomando.
COD. 37/310 €. 360.00

Navigatore Mio268 Sw Italia
SD 256Mb SW Italiano.
COD. RC268I €. 422,00

Kit telecamera 1/3CMOS waterproof a colori + ricevitore senza fili con illuminatore IR operante sulla banda dei 2.4GHz.il kit include telec., ricev. a 4 canali e alimentatori.
COD. 37/150 €. 115,00

Multimetro digitale
con ingressi protetti
COD. TE/6300E €. 48,00

Tutti i prezzi si intendono IVA inclusa.

Descrizione strutturale

La descrizione strutturale rappresenta il circuito in termini di blocchi logici (quali porte, flip-flop, memorie, registri, eccetera) presenti nella libreria e collegati tra di loro a costituire la *lista segnali* (*netlist*). Le informazioni supportate da questa descrizione riguardano i ritardi della logica e l'area occupata. Sono esclusi i ritardi dovuti ai collegamenti.

Figura 12 *Form per la creazione del sorgente*

Figura 13 *Inizializzazione effettuata*

Figura 14 *Scelta del tipo di file*

Implementazione

La fase di implementazione traduce la descrizione in termini di gates logici in una descrizione fisica, che si traduce infine in un file di programmazione da scaricare nel dispositivo.
Il flusso per CPLD (semplificato rispetto a quello per FPGA) contiene i seguenti step (come da figura 16b):

- Translate (cioè traduzione con blocchi pre-configurati)
- Fit (mappatura della logica nelle macrocelle)

Il procedimento fa uso dei vincoli implementativi

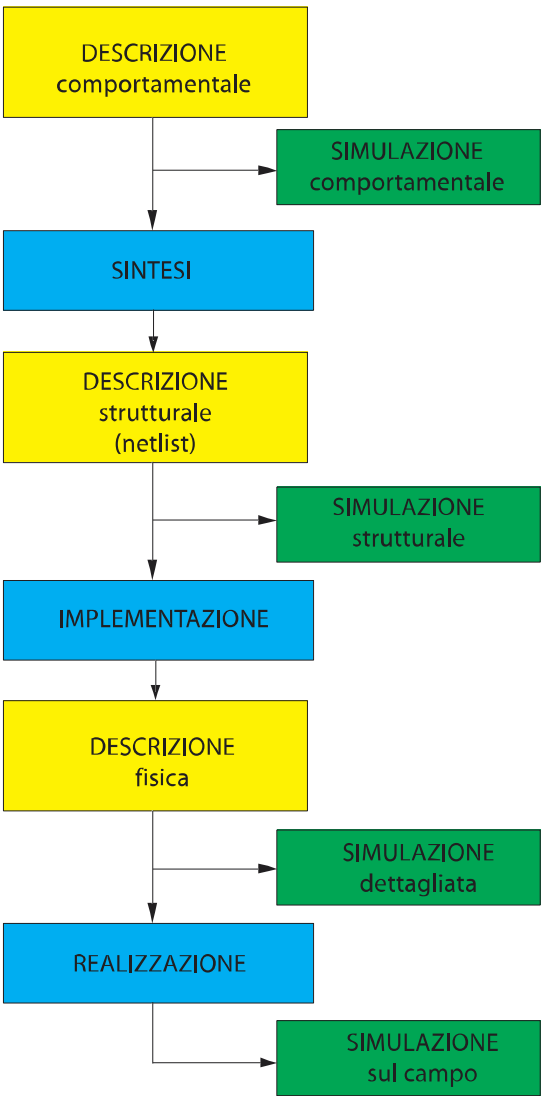
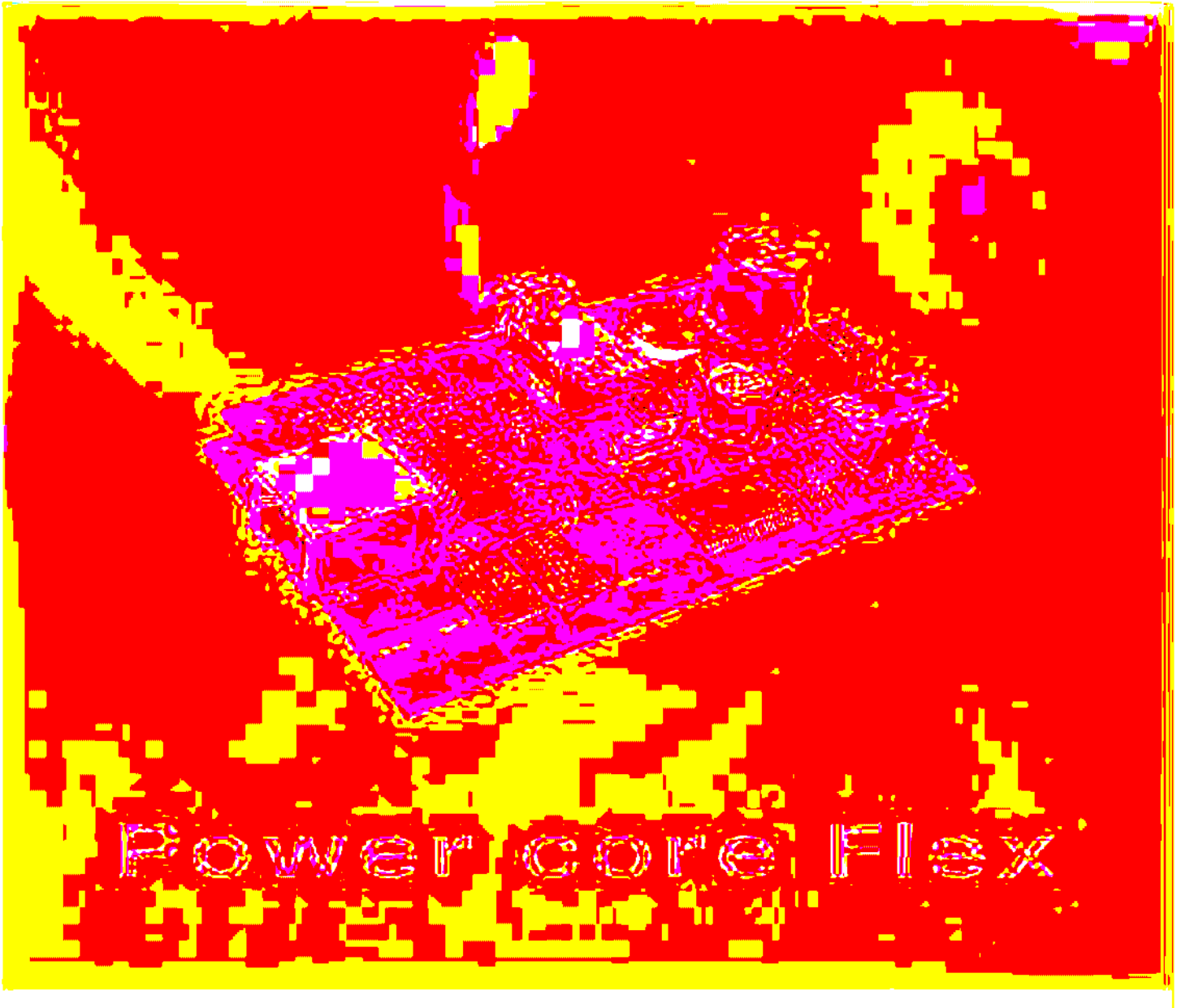


Figura 15 *Fasi del flusso di progetto*



RABBIT Semiconductor

CONNETTI LE TUE IDEE CON LE NOSTRE

piccoli componenti, per grandi progetti: i tuoi.

COMPREL an esprinet company

Comprel Componenti Competenti

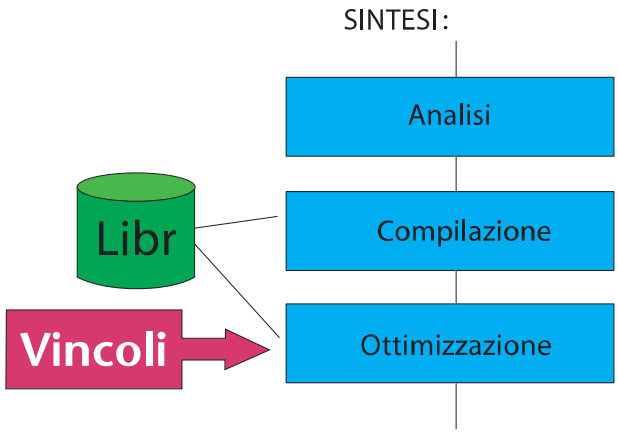


Figura 16a *Passi della Sintesi*

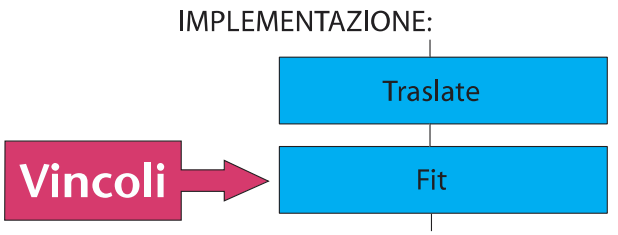


Figura 16b *Passi relativi all'implementazione*

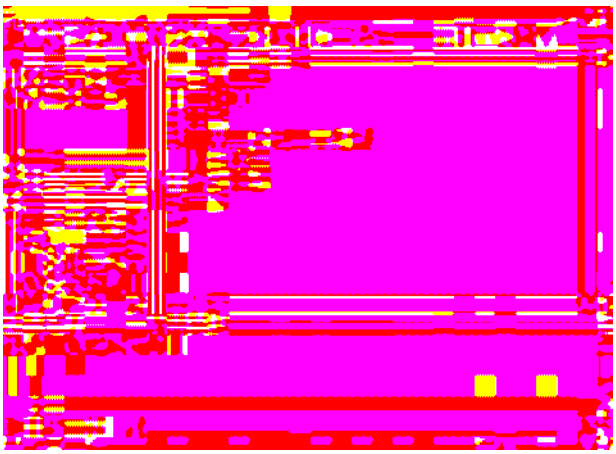


Figura 17 *Sintesi andata a buon fine*

Figura 18 *Sintesi non corretta*

così come sono stati indicati nel file di *constraint* UCF, come ad esempio il *pin lock*.

Descrizione fisica

La descrizione fisica è quella che rappresenta il circuito in modo molto prossimo a quella reale, in quanto contiene le informazioni sui ritardi nella logica, sui ritardi introdotti dai collegamenti, sull'area occupata e il dettaglio sull'ubicazione di ogni singola porta logica e dei pin di I/O.

Realizzazione

È la fase finale del procedimento; consiste nel trasferimento del codice di programmazione sul componente.

Dettaglio delle fasi

Chiusa la parentesi sul flusso di progetto, riprendiamo ora il nostro circuito. Si era già effettuata la descrizione comportamentale con il codice sorgente; procediamo con la Sintesi.

Fase di Sintesi

Per far eseguire la fase di Sintesi, clicchiamo su *Synthesize* nella finestra dei processi, a sinistra dello schermo.

Una volta terminata questa fase vedremo dei simboli di "spunta" di colore verde su tutti gli step che sono andati a buon fine (figura 17).

Figura 19 *Implementazione ok*

Se vi fossero degli errori vedremmo

Figura 20 *Generazione file JTAG*

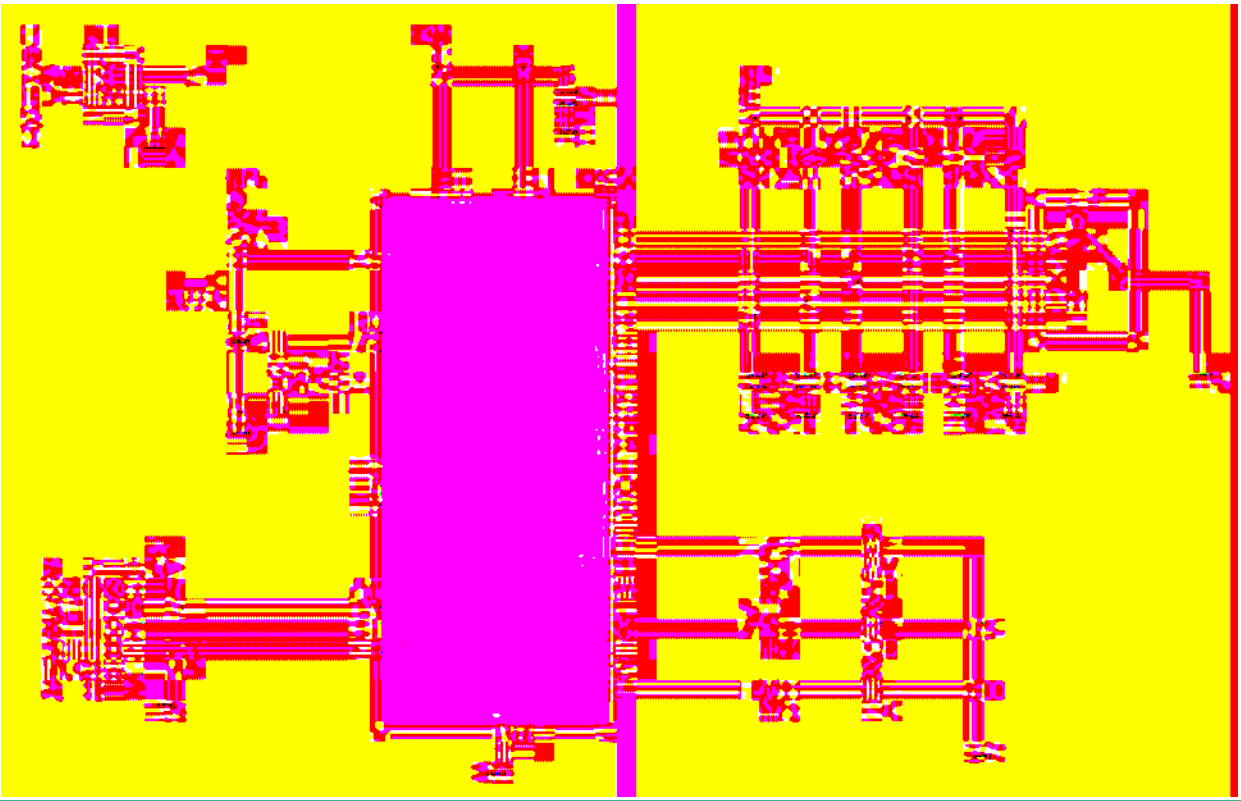


Figura 21 Schema elettrico per l'esempio 2 (encoder)

comparire delle crocette rosse (come da figura 18).
La sintesi genera un file di *Report*, visualizzabile cliccando su *View Synthesis Report*, in cui vengono inserite tutte le informazioni relative alle risorse utilizzate nei sottoprocessi di *Analisi*, *Compilazione* e *Ottimizzazione*.
In particolare, nella sezione riassuntiva in fondo al report troviamo che il tool di sintesi ha riconosciuto la presenza di un registro da 1 bit, a cui ha associato un flip-flop di tipo D.
Per quanto riguarda invece gli ingressi e le uscite del modulo, possiamo vedere che sono stati allocati due buffers di ingresso (*IBUF*) e un buffer di uscita (*OBUF*).

Fase di Implementazione

Procediamo ora con la fase di Implementazione, cliccando su *Implement Design*, sempre nella finestra dei processi.
Al termine di questa fase (figura 19) potremo visualizzare il file di report relativo, che ci mostrerà le operazioni compiute.
In particolare, ci può interessare il sommario delle risorse utilizzate, cioè quante macrocelle sono state impegnate, quanti termini di prodotto, quanti function blocks, quanti pin di ingresso/uscita, eccetera.

Elenco componenti ESEMPIO 2

Sigla	Valore
R01÷R09	10 KΩ 1/4 W
R10÷R12	390 Ω 1/4 W
C01÷C07	0,1 µF 50 V ceramico
C08	1 µF 20 V elettrolitico
SL01÷SL03	Diodo LED
IC01	CPLD XC9536-15PC44C (package 44 PIN PLCC)
J01	Connettore di alimentazione da stampato
J02	Connettore lineare a striscia, 6 pin
SW01	Commutatore rotativo 1 via 6 posizioni

Generazione e trasferimento del file di programmazione

Per generare il file di programmazione occorre dare un doppio click su *Generate Programming File*. Dopodichè si potrà eseguire *Generate JTAG File*.
Quest'ultima operazione (figura 20) richiede

attenzione in quanto bisogna ricordarsi di aver collegato l'interfaccia JTAG al circuito e di aver acceso l' alimentatore.
A questo punto, dall'ambiente Project Navigator si richiama l'applicativo *IMPACT*, il quale trasferisce il file di programmazione sul dispositivo reale. La descrizione dettagliata di quest'ultima fase la possiamo trovare nel numero 228 di Fare Elettronica (Giugno 2004).

LA SIMULAZIONE

Come già visto, nel flusso di progetto è prevista in vari punti una sessione di simulazione. Ad esempio, la *simulazione comportamentale* viene effettuata prima della sintesi per verificare che la logica che è stata creata sia corretta.
Questo permette al progettista di rilevare gli eventuali "bugs" nascosti del progetto prima di spendere tempo per la *sintesi* o l'*implementazione*.
La simulazione non è strettamente necessaria, ma è essenziale se si vuole avere la certezza del funzionamento di un circuito logico in tutte le sue parti. Infatti, anche la codifica più perfetta che si possa immaginare può contenere difetti, se pur lievi.
Una volta che il progetto è stato completamente "simulato", anche con l'introduzione dei ritardi reali, si può essere ragionevolmente sicuri che la CPLD si comporterà come da specifica.

INGRESSI						USCITE		
I023 (I5)	I024 (I4)	I025 (I3)	I026 (I2)	I027 (I1)	I028 (I0)	I01 (O2)	I05 (O1)	I09 (O0)
1	1	1	1	1	1	0	0	0
1	1	1	1	1	0	0	0	1
1	1	1	1	0	1	0	1	0
1	1	1	0	1	1	0	1	1
1	1	0	1	1	1	1	0	0
1	0	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	0

Tabella 2 *Funzione di codifica*

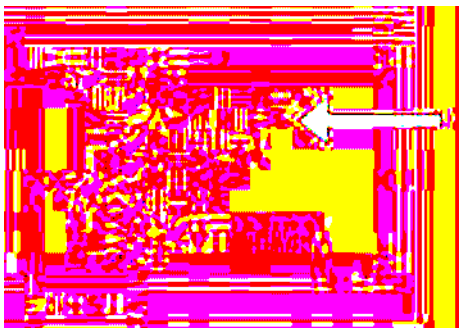


Figura 22 *Visualizzazione della schematica del progetto*

MODULO 3

In questo modulo approfondiamo le caratteristiche del tool di sintesi e realizziamo un paio di esperimenti di logica combinatoria.

ESEMPIO 2: ENCODER ROTATIVO

Schema elettrico

Anche per questo secondo esempio valgono le considerazioni fatte in precedenza. Data la semplicità circuitale ne presentiamo solo lo schema elettrico (figura 21) supponendo di realizzarlo su una basetta millefori e di montare la CPLD su zoccolo.
Il componente scelto è ancora la XC9536, ma si può utilizzare anche la XC9572, che è pin-compatibile ed ha una capacità superiore. Possiede infatti 72 macrocelle, con 1600 gates utilizzabili.
I *Function Blocks*, di tipo 36V18, consentono di avere fino a 90 termini di prodotto per ognuna delle 18 macrocelle presenti. L'architettura interna è un sottoinsieme di quella di figura 1.
Questo esempio ha, di particolare, il fatto che non utilizza clock di ingresso.
Si tratta di un encoder per switch a 6 posizioni, quindi di una rete puramente combinatoria. Le resistenze di pull-up R04÷R09 servono a polarizzare gli ingressi a livello high.
A seconda di come viene posizionata la manopola dello switch, uno solo alla volta dei pin di ingresso viene posto a livello low. La CPLD interpreta questa situazione e restituisce in uscita un codice binario a 3 bit che codifica la posizione del commutatore. La funzione realizzata è riportata in tabella 2.

Codice sorgente

Il codice sorgente realizza una tavola di corrispondenza tra gli ingressi IO÷I5 e le uscite O0÷O2 per mezzo del costrutto case, come di seguito riportato:

```
module
top(reset,I0,I1,I2,I3,I4,I5,O0,O1,O2);
input   reset,I0,I1,I2,I3,I4,I5;
output  O0,O1,O2;
reg [2:0] data_out;

// tabella di codifica
always @ (I0 or I1 or I2 or I3 or I4 or I5
or reset)
  begin
    if (!reset)
      data_out =  3'b000;  // tutto spento
al reset
    else
    case({ I5,I4,I3,I2,I1,I0 })
     //   INGRESSI            USCITE
     //                      O2 O1 O0
     6'b111110 : data_out =  3'b001;  // 1
     6'b111101 : data_out =  3'b010;  // 2
     6'b111011 : data_out =  3'b011;  // 3
     6'b110111 : data_out =  3'b100;  // 4
     6'b101111 : data_out =  3'b101;  // 5
     6'b011111 : data_out =  3'b110;  // 6
     default : data_out =  3'b000;
    endcase
end
   assign {O2,O1,O0} = data_out;
endmodule
```

FILE DI CONSTRAINT

```
net  RESET   LOC="P39";
net  I0      LOC="P44";
net  I1      LOC="P43";
net  I2      LOC="P38";
net  I3      LOC="P37";
net  I4      LOC="P36";
net  I5      LOC="P35";
net  O0      LOC="P13";
net  O1      LOC="P8";
net  O2      LOC="P1";
```

Nel report del processo di *Fitter* potremo vedere le equazioni logiche in cui la nostra tabella è stata tradotta.

Schematica equivalente

Il tool ISE è in grado di generare la *Schematica Equivalente* al codice sorgente. Per vederla dobbiamo scegliere l'opzione *View RTL Schematic*, nella finestra dei processi (figura 22).

Si aprirà una nuova finestra di visualizzazione contenente lo schema dell'encoder. A partire dal top-level, possiamo scendere nella gerarchia cliccando con il mouse sui blocchi di cui è composto, fino ad arrivare alle porte logiche elementari (figure 23 a, b).

Chip Viewer

Un'altra caratteristica molto interessante fornita dal tool di sviluppo è il cosiddetto ***Chip Viewer***, che consente la visualizzazione delle risorse interne reali che sono state utilizzate nel progetto.
Per attivarlo clicchiamo su *View Fitted Design* (figura 24). Il risultato è visibile nella figura 25.

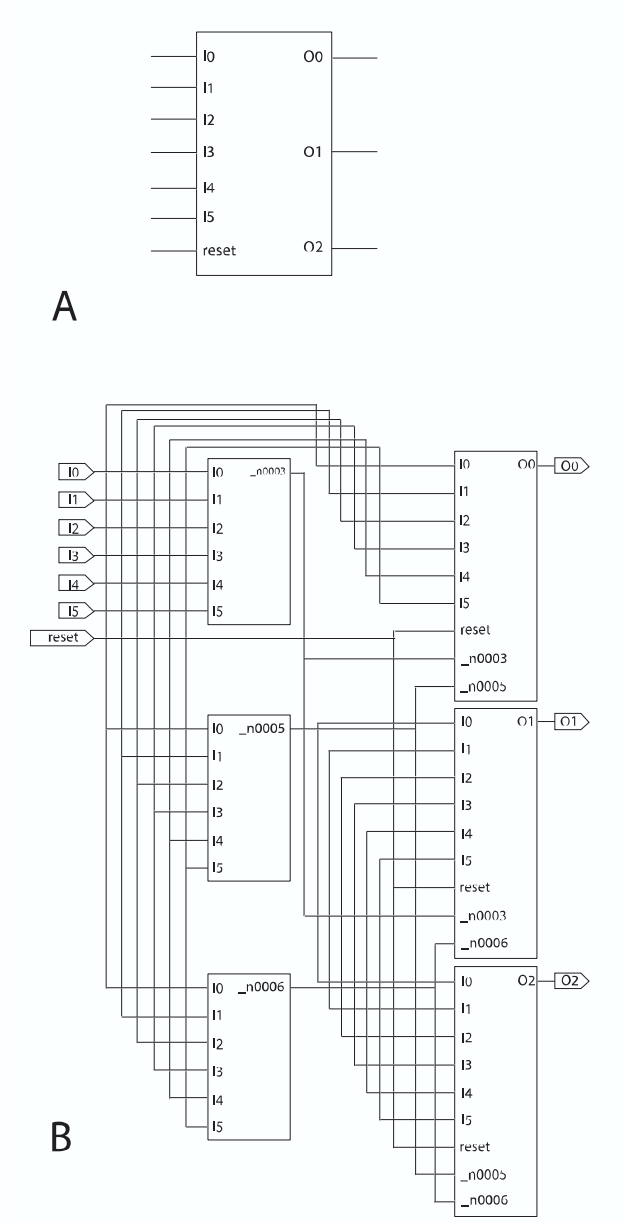


Figura 23 a, b *Descrizione della schematica gerarchica del progetto*

Anche in questo caso è molto comodo il fatto di poter espandere la struttura in forma gerarchica. È possibile visualizzare anche le connessioni di ingresso e uscita (evidenziate in rosso e giallo rispettivamente).

ESEMPIO 3: PILOTARE UN DISPLAY A 7 SEGMENTI

Schema elettrico

Con poche modifiche si può trasformare lo schema dell'esempio 2 in un pratico sistema di visualizzazione della posizione del commutatore per mezzo di un display a 7 segmenti. Lo schema così modificato è descritto in figura 27.

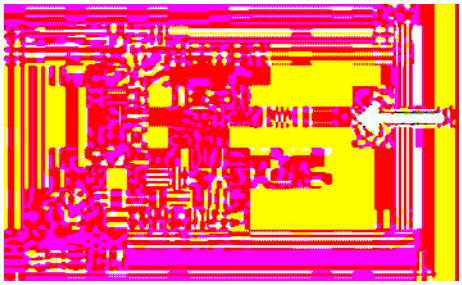


Figura 24 Attivazione di Chip Viewer

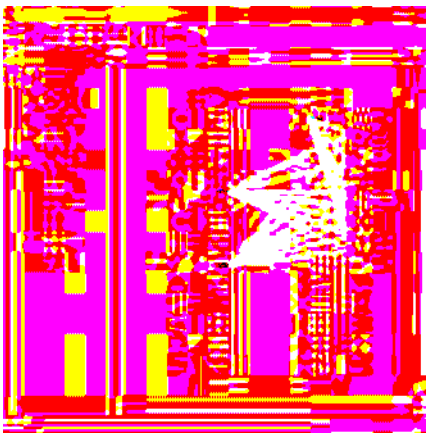


Figura 25 Visualizzazione delle risorse impiegate

Figura 26 Dettaglio della macrocella 11, entro il Function block FB1, relativa all'uscita O0 (pin 13)

Codice sorgente

Possiamo notare che il codice sorgente è molto simile a quello dell'esempio precedente; infatti è sufficiente inserire i pin di uscita relativi ai segmenti del display IC02 e modificare il costrutto case in modo da prevedere l'accensione dei segmenti così da formare le cifre 1÷6 in rapporto alla posizione del commutatore, come di seguito riportato:

```
module top (reset,I0,I1,I2,I3,I4,I5,SEG_A,
   SEG_B,SEG_C,SEG_D,SEG_E,SEG_F,SEG_G,SEG_DP);

input    reset,I0,I1,I2,I3,I4,I5;
output   SEG_A,SEG_B,SEG_C,SEG_D,SEG_E,SEG_F,
         SEG_G,SEG_DP;

reg [7:0] data_out;

// tabella di corrispondenza
always @ (I0 or I1 or I2 or I3 or I4 or I5
or reset)
  begin
    if (!reset)
      // tutto spento al reset
      data_out =  8'b11111111;
    else
      case({ I5,I4,I3,I2,I1,I0 })

// INGRESSI                USCITE
//                        A,B,C,D,E,F,G,DP
   6'b111110 : data_out = 8'b10011111; // 1
   6'b111101 : data_out = 8'b00100101; // 2
   6'b111011 : data_out = 8'b00001101; // 3
   6'b110111 : data_out = 8'b11011001; // 4
   6'b101111 : data_out = 8'b01001001; // 5
   6'b011111 : data_out = 8'b01000011; // 6
     default : data_out = 8'b11111111;
   endcase
end

assign  {SEG_A,SEG_B,SEG_C,SEG_D,SEG_E,SEG_F,
         SEG_G,SEG_DP} = data_out;

endmodule
```

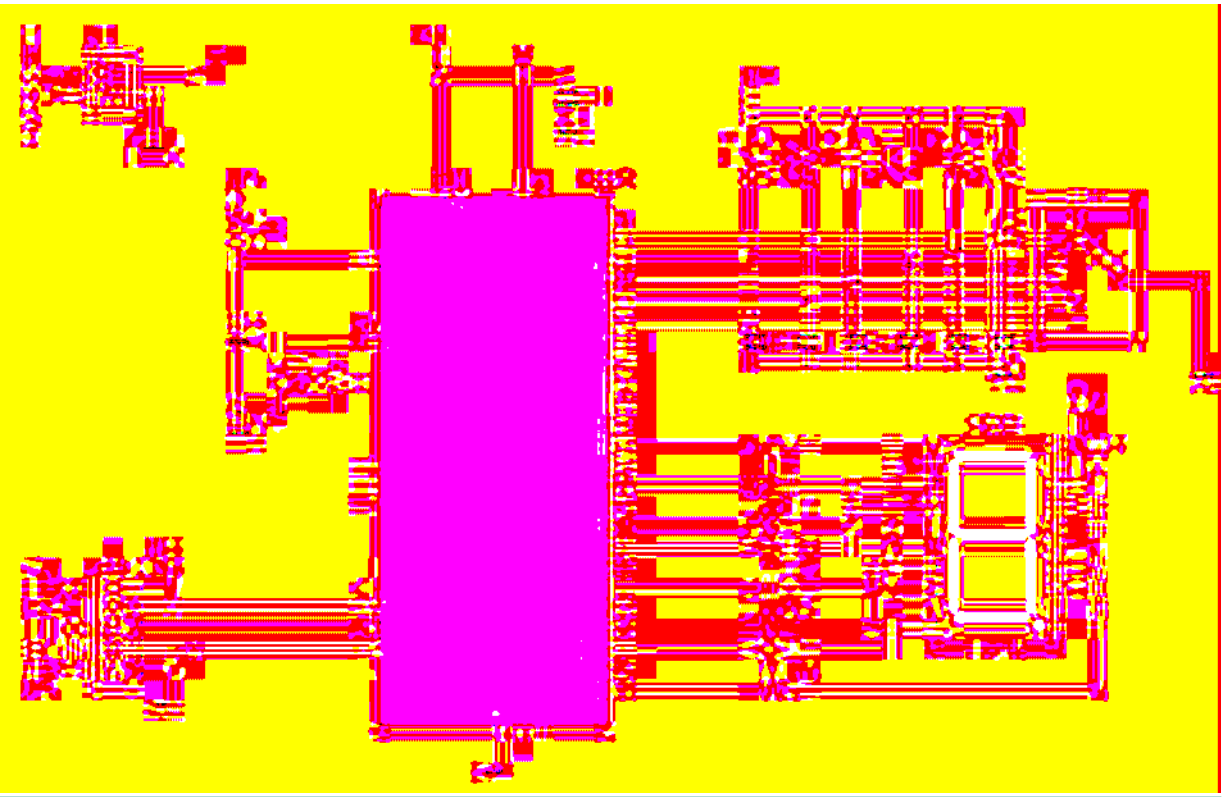


Figura 27 *Schema elettrico per l'esempio 3*

FILE DI CONSTRAINT

```
net RESET     LOC="P39";
net   I0      LOC="P44";
net   I1      LOC="P43";
net   I2      LOC="P38";
net   I3      LOC="P37";
net   I4      LOC="P36";
net   I5      LOC="P35";
net  SEG_A    LOC="P24";
net  SEG_B    LOC="P20";
net  SEG_C    LOC="P18";
net  SEG_D    LOC="P13";
net  SEG_E    LOC="P11";
net  SEG_F    LOC="P8";
net  SEG_G    LOC="P3";
net  SEG_DP   LOC="P1";
```

In questo caso il report del processo di Fitter ci riporta percentuali di utilizzo delle risorse interne del chip ovviamente un pò più elevate rispetto al caso precedente ed equazioni logiche (per le uscite) molto più complesse.

PROSSIMAMENTE

Nella prossima puntata presenteremo il circuito-base didattico che servirà da supporto per le successive sperimentazioni.

Elenco componenti ESEMPIO 3	
Sigla	**Valore**
R01÷R09	10 KΩ 1/4 W
R10÷R17	390 Ω 1/4 W
C01÷C07	0,1 µF 50 V ceramico
C08	1 µF 20 V elettrolitico
IC01	CPLD XC9536-15PC44C (package 44 PIN PLCC)
IC02	Display ad anodo comune Fairchild MAN3910A
J01	Connettore di alimentazione da stampato
J02	Connettore lineare a striscia
SW01	Commutatore rotativo, 1 via 6 posizioni

Si tratta di un circuito molto semplice e realizzabile con una spesa modesta, che potrà essere espanso (come faremo effettivamente) con l'aggiunta di moduli di interfaccia per svolgere specifiche esercitazioni.
Vedremo poi i vari tipi di operatori Verilog insieme a numerosi esempi applicativi.

Leader nella progettazione e produzione di circuiti ibridi con tecnologia in film spesso

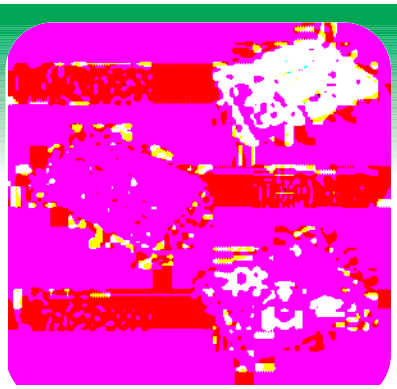
Modello	Alimentazione	Sensibilità RF/Potenza trasmissione	Frequenza (xxx)	Velocità trasmissione	Descrizione
RR30-xxx	5Vdc/2,5mA	-105dBm	300-450MHz	4,8Kbps	Coppia di moduli AM di dimensioni estremamente compatte (TX = 17,8 x 10,2mm, RX = 25,4 x 8,9mm), omologati I-ETS 300-220, idonei per applicazioni di controllo remoto
RT4-xxx	2÷14Vdc/4mA	+7dBm	303,8-433,92MHZ	4Kbps	

Modello	Alimentazione	Sensibilità RF	Frequenza (xxx)	Velocità trasmissione	Descrizione
RR18-xxx	3Vdc/70mA	-96dBm	433,92MHz	4,8Kbps	Ricevitore AM superigenerativo a basso consumo, con filtro saw in ingresso e banda stretta

Modello	Alimentazione	Potenza trasmissione	Frequenza (xxx)	Velocità trasmissione	Descrizione
RTQ4-xxx	2÷5,5Vdc/12mA	+7dBm	433,92MHz 868,35MHz 915MHz	9,6Kbps	Trasmettitori AM con oscillatore al quarzo, dimensioni compatte (17,78 x 10,16 mm) e pin out compatibile con il modello RT4

Modello	Alimentazione	Potenza trasmissione	Frequenza (xxx)	Velocità trasmissione	Descrizione
RTFQ4-xxx	2÷5,5Vdc/12mA	+7dBm	433,92MHz 868,35MHz 915MHz	9,6Kbps	Trasmettitore FM con oscillatore al quarzo, dimensioni compatte (17,78 x 10,16 mm) e pin out compatibile con il modello RT4

Modello	Alimentazione	Sensibilità RF	Frequenza	Velocità trasmissione	Descrizione
RXQ1-XXX	2.7÷5,25Vdc/12mA	-100dBm	433,92MHz 434,33MHz	20Kbps	Transceiver a 2 canali
TRXQ1-XXX	2.7÷5,25Vdc/12mA	-100dBm	433,92MHz 434,33MHz	20Kbps	Transceiver a 2 canali con encoder/decoder
RXQ2-xxx	2÷3,6Vdc	-100dBm	433,92MHz	38,4Kbps	Transceivers multicanale per trasmissione dati veloci e sicure Completo d'interfaccia RS-232
	2÷3,6Vdc	-100dBm	868,35MHz	38,4Kbps	



Maggiori informazioni sono disponibili nel sito **www.telecontrolli.com**,
campioni disponibili per vendita on-line tramite **www.elettroshop.com** nella sezione "Moduli RF".

Decima parte
n° 244 - Ottobre 2005
Switching flyback multi-uscita

Undicesima parte
n° 245 - Novembre 2005
Analisi di uno switching off-line (parte I)

Dodicesima parte
n° 246 - Dicembre 2005
Analisi di uno switching off-line (parte II)

Alimentatori switching:

In questa puntata cominceremo l'analisi di un alimentatore switching da PC, che completeremo nella prossima, ed ultima, puntata. Nonostante la trattazione puramente teorica, non mancheranno utili precisazioni pratiche sul funzionamento di questi "mostri" di potenza, che mi auguro possano fornire la base per ulteriori approfondimenti da parte vostra.

PREMESSA

Concedetemi di ripetermi: in questa puntata esamineremo (smontandolo!) un alimentatore switching PC, alimentato da rete, quindi prima di imitarmi...

Attenzione

- **Non mettete le mani in uno switching da rete se non conoscete i rudimenti dell'elettricità!**
- **Usate sempre un interruttore doppio per disconnettere l'alimentazione!**
- **Attendete sempre almeno 60/120 secondi, dopo la disconnessione dell'alimentazione, per permettere ai condensatori di scaricarsi! (misurate con il tester in caso di dubbio).**

Se comunque vorrete osare, lo fate a vostro rischio e pericolo, e sia lo scrivente che FE declinano qualsiasi responsabilità ;-).

L'ALIMENTATORE

L'alimentatore in questione è stato smontato dal mio PC, in quanto aveva la "bella" abitudine di generare una tensione troppo elevata sui 3,3V... con rischio di cottura dei miei preziosissimi componenti!

Si tratta di un discreto 300W, con PFC attivo (vedremo nella prossima puntata cosa questo significhi), ovviamente Made in China. Per chi non lo sapesse, un alimentatore PC nel "nuovo" formato ATX deve fornire le tensioni di: **+3,3V +5V -5V +12V -12V.**

Di queste le più importanti sono quelle positive, cioè la +3,3V, la +5V e la +12V. Le tensioni negative -5V e -12V sono un retaggio dei primi alimentatori per PC, e probabilmente non sono più usate.

Vediamo quante cose possiamo capire da una semplice etichetta (figura 1).

Innanzitutto osservate i dati di targa: notate come la tensione di ingresso specificata sia 230VAC. Se avete qualche vecchio alimentatore, potrete trovare anche segnalata una tensione di ingresso di 110-230VAC, ma in questo caso in genere è presente sul retro un interruttore che commuta da 110VAC a 230VAC. Vedremo più avanti il perchè di questa soluzione alternativa. Come secondo dato, notate come la corrente dichiarata sia pari a ben 5A. Se ricordate la semplice formula della potenza, risulterebbe una potenza assorbita pari a: P_{ASS} = 230 * 5 = 1150W!

A fronte di soli 300W erogati!? Come si spiega un'efficienza così bassa?

Si spiega perchè in realtà i 5A dichiarati rappresentano il valore di assorbimento per dimensionare i cablaggi (cavo, presa nel muro, eccetera) e non è la corrente efficace assorbita.

Noterete che questa peculiarità, cioè dichiarare una corrente di ingresso molto maggiore di quella effettivamente assorbita, è comune a tutti gli alimentatori switching, per conferma

Analisi di uno switching off-line (parte I)

di Romano Bernarducci
r.bernarducci@farelettronica.com

date un'occhiata ai caricabatterie dei vostri notebook o cellulari.
Notate inoltre, a conferma di quanto detto prima, che la corrente maggiore viene erogata sulle tensioni positive, e precisamente 20A a +3,3V, 30A a +5V e 15A a +12V, mentre sulle tensioni negative abbiamo solo 0,3A a -5V e 0,8A a -12V.
Esiste poi una tensione, per così dire, supplementare, indicata con la sigla $+5V_{SB}$, per una corrente massima di 1,5A. **É questa la tensione di servizio, sempre presente anche quando il PC è spento.** Per motivi che sarebbe lungo spiegare (implementazione del Wake-On-Lan, dello spegnimento automatico ed altro), l'alimentatore in standby fornisce comunque questi +5V, in maniera analoga al TV di casa quando resta con la lucina rossa accesa.
Quando l'utente preme il pulsante di accensione (non è più un interruttore come nei vetusti PC), l'alimentatore viene attivato a piena potenza per mezzo del segnale REMOTE/PS-ON, indicato secondo da destra nell'etichetta di figura 1.

Quindi, per chi volesse utilizzare un alimentatore PC ATX per il proprio laboratorio, per accenderlo basta che colleghi **il filo PS-ON (filo grigio) a massa (filo nero).**
Continuando l'analisi dell'etichetta, notate che per i +5V, è definita sia la corrente massima (30A), che la corrente minima (2,5A), a differenza di tutte le altre uscite.
Facciamo un capitombolo indietro alle primissime puntate... ricordate che se il carico dello switching boost (senza controller) era inferiore al carico minimo, addio regolazione… qui avviene più o meno la stessa cosa!
Ora è ovvio che in questo caso il controller c'è (e sicuramente avrà la controreazione sui +5V), tuttavia **essendo il nostro switching un multi-uscita, il funzionamento "intermittente" sui +5V, dovuto all'assenza di carico, compromette la stabilità di tutte le altre uscite.**
Quindi, se proprio volete usarlo per il vostro laboratorio, abbiate l'accortezza di mettere un carico fisso sui +5V che assorba almeno 2,5A, ad esempio una resistenza pari a $R = V / I = 5 / 2,5 = 2\Omega$, di potenza pari almeno a $P = I^2 * R = 2,5^2 * 2 = 12,5W$ montata su adeguato dissipatore!
Non temete per i 2,5A "persi" in calore sulla resistenza, ne avete ancora a disposizione ben 27,5A!
Per finire lo strazio dell'etichetta, nella metà in basso trovate una serie di strani simboli: sono le sigle degli enti certificatori nazionali riguardanti la rispondenza a norme di Sicurezza Elettrica, la sigla CE necessaria per la vendita di qualsiasi prodotto nella Comunità Europea, e, dulcis in fundo, alcune scritte di ammonimento a non

Figura 1 L'etichetta dell'alimentatore PC

Figura 2 Esterno dell'alimentatore PC

aprire il contenitore...a buon intenditor...

Diamo ora uno sguardo all'esterno dell'alimentatore, cioè l'unica parte visibile quando questo è montato all'interno del PC, vedi figura 2.

In basso a destra è visibile l'interruttore di alimentazione, cioè quello che ha il compito di spegnere totalmente l'alimentatore, in posizione 0. **In caso contrario l'alimentatore fornirà comunque i +5V$_{SB}$, in attesa che l'utente prema il pulsante di accensione sul frontale del PC.**
In alto a destra trovate il classico connettore di alimentazione a tre poli, dove il pin centrale, più sporgente, corrisponde alla terra. Potete provate con il tester, in misura di continuità, tra questo pin e la carcassa, dovrebbe suonare, altrimenti sono guai: l'alimentatore è non solo o non necessariamente rotto, ma è PERICOLOSO.
A sinistra è chiaramente visibile la griglia della ventola di raffreddamento, sempre necessaria in alimentatori di queste dimensioni. Infatti, se supponiamo un'efficienza del 75%, relativamente ai 300W erogati, ci saranno almeno 100W, pari al restante 25%, dissipati in calore e **proibitivi da dissipare per convezione naturale (almeno in questi piccoli contenitori).**
Mentre nei vecchi alimentatori PC non ATX, la ventola era spesso alimentata a 220VAC, in quanto l'interruttore di accensione spegneva comunque tutto, nei nuovi ATX non sarebbe possibile in quanto i 230VAC sono sempre presenti (servono infatti per generare la tensione di Stand-By), e la ventola girerebbe sempre. **Si preferisce quindi alimentarla con i +12V che si rendono disponibili al momento dell'accensione.**
Nell'alimentatore in oggetto la ventola è termostatica, cioè la sua velocità di rotazione dipende dalla temperatura raggiunta all'interno dell'alimentatore, a sua volta dipendente dal carico ad esso collegato. In teoria questo sistema garantisce che in PC "normali", con pochi componenti interni e processori "normali", per i quali l'assorbimento è medio/basso, il rumore prodotto dalla rotazione della ventola sia proporzionalmente ridotto...che poi questo sistema sia realmente efficace è un altro discorso, spesso è solo una nota di marketing.

Diamo uno sguardo all'interno, figura 3.
Ho evidenziato con riquadri colorati alcuni blocchi funzionali, esaminiamoli.
In rosso ho indicato lo stadio collegato direttamente alla presa di alimentazione già vista, che contiene il FILTRO DI INGRESSO.
In blu ho indicato il circuito del **PFC attivo**, Power Factor Corrector, cioè il Correttore del Fattore di Potenza (vedremo meglio nella prossima puntata). L'indicazione attivo significa che è realizzato tramite semiconduttori, nello specifico, un circuito integrato della ST.
In verde lo schedino del controllo termostatico della ventola, che come vedete è montato direttamente sul dissipatore dello stadio di uscita, in modo da rispondere meglio alle relative variazioni di temperatura.

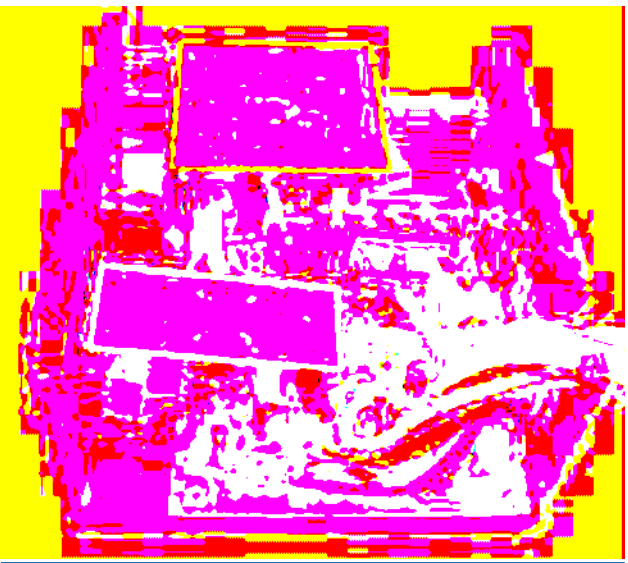


Figura 3 *Vista interna dell'alimentatore PC*

Figura 4 *Particolare del filtro di ingresso*

Una serie completa di scatole di montaggio bi-tegrale strutture per rete GSM.

FUTURA ELETTRONICA

Notate come i due dissipatori (quello in secondo piano è per gli interruttori sul primario) creino una sorta di canale attraversato dall'aria pompata dalla ventola verso l'esterno, e come in mezzo a tale canale sia disposto anche il trasformatore principale, appena visibile in figura 3 con la sigla ATERL35T03.
Prima di proseguire, un avviso: questi alimentatori, pur avendo potenze e caratteristiche "fantascientifiche", sono in vendita per pochi euro (20-30), quindi aspettatevi di trovarvi utilizzate le più scadenti soluzioni costruttive. Ad esempio notate i grumi di colla a caldo (sic) che tenevano in posizione lo schermo isolante di plastica sopra il PCB del filtro di ingresso...

FILTRO DI INGRESSO

Proseguiamo nella vivisezione dell'alimentatore, smontando lo schedino del FILTRO DI INGRESSO. Per far questo ho dovuto dissaldare due fili ed il connettore di alimentazione, vedi figura 4. In figura 5 ho riportato lo schema elettrico corrispondente.

Partendo dalla sinistra, incontriamo immediatamente il condensatore CX0, saldato direttamente sui pin del connettore a vaschetta (vedi figura 4). Notate come questo condensatore, e anche gli altri presenti nello schema di figura 5, abbia stampigliato sul corpo varie sigle di enti normatori, europei e mondiali, operanti nel settore della sicurezza elettrica: ne vedremo i motivi nel prossimo paragrafo.
Notate pure che sul corpo di CX0 è presente una piccola sigla "**X2**": questa caratterizza un particolare tipo di condensatori con **caratteristiche tali da poter essere sottoposti, in sicurezza, direttamente alla tensione di rete.**
Per questo motivo **è importante che questi componenti non vengano mai sostituiti con condensatori "normali"**, anche se di pari valore e tensione di lavoro.
Quale è la funzione di un filtro di ingresso?
La risposta è duplice...in primo luogo evita che i disturbi sempre presenti sulla rete elettrica, possano entrare nell'alimentatore danneggiandolo o provocandone l'errato funzionamento. In secondo luogo, il filtro impedisce il contrario, e cioè che i disturbi generati dallo switching durante il suo funzionamento entrino in rete, "sporcandola".
Esiste in proposito una estesa normativa sulle emissioni massime consentite per i dispositivi elettrici connessi alla rete elettrica.

I CONDENSATORI SPECIALI "X" E "Y"

La differenza fondamentale tra i condensatori "speciali" X e Y, è che **i primi (X) sono usati nei punti in cui la loro eventuale rottura non può comunque causare uno shock elettrico, mentre i secondi (Y) nei punti in cui la loro rottura potrebbe causare shock elettrico.**
Da questo punto di vista è ovvio che i condensatori tipo Y devono essere "migliori" di quelli tipo "X", che a loro volta devono comunque essere migliori dei condensatori standard, non marchiati.
I tipo X potrebbero infatti sopportare la tensione di rete, compresi i disturbi e le sovratensioni sovrapposte, anche per tempi indefiniti: dalla figura 5, infatti, tutti i condensatori sono

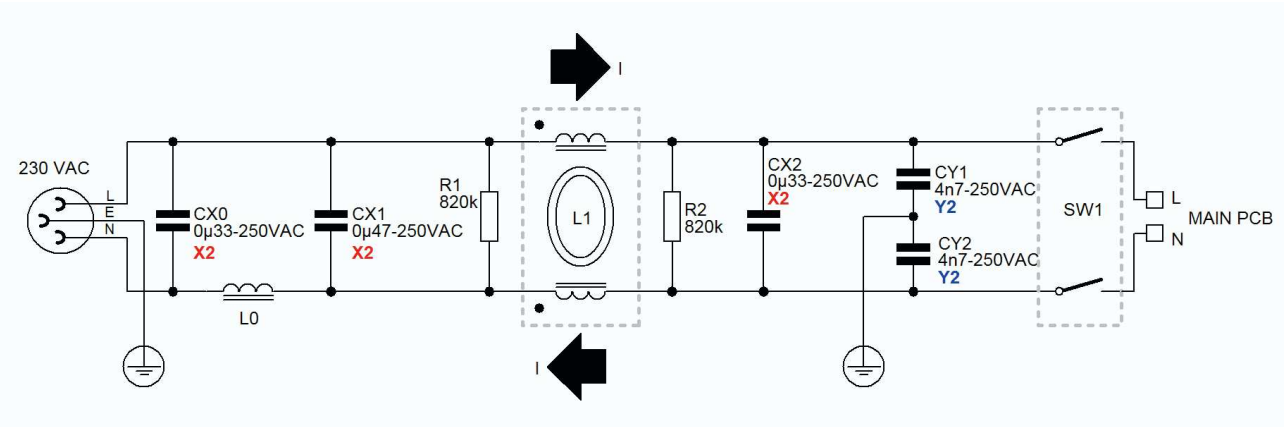


Figura 5 Schema elettrico del filtro di ingresso

sempre connessi alla rete, almeno finchè il cavo è collegato alla presa, **anche quando l'interruttore di alimentazione è spento!**
L'eventuale rottura di un condensatore "X", deve comunque essere "sicura", cioè non dovrebbe mai causare incendi, esplosioni, eccetera. Per questo motivo a volte nei loro datasheet si indica la proprietà "dielettrico autoestinguente".
Nello schema di figura 5, **tutti i condensatori sottoposti alla tensione di rete sono almeno di tipo "X"**, tranne i due all'estrema destra che sono invece di **tipo "Y", perchè sono connessi tra la rete e la terra.** É evidente infatti, che una loro rottura, ad esempio un cortocircuito interno, potrebbe collegare un filo di rete (L o N), alla terra elettrica, cioè allo chassis dell'apparato, con conseguente pericolo per l'utente.
Esistono tre classi per i condensatori tipo "X": **X1, X2 e X3** e quattro classi per i condensatori di tipo "Y": **Y1, Y2, Y3 e Y4**, tuttavia negli apparati commerciali vedrete di norma solo le **X2** e **Y2**, e raramente le **X1** e **Y1** in apparati di alta qualità, ad esempio per ambienti industriali. Le classi inferiori non sono sufficienti per le normative attuali e, sinceramente, non mi è mai capitato di trovarne e nemmeno ne comprendo l'utilità.
La classe indica la resistenza alle sovratensioni per le prime due classi (vedi tabella 1).

A volte si possono trovare condensatori marchiati con entrambe le sigle "X1" e "Y2", in quanto dalla tabella è ovvio che **un condensatore di tipo Y può sempre essere usato al posto di un tipo X di classe equivalente.** Lo svantaggio in questo caso, è legato alle dimensioni ed al costo, entrambi maggiori nei tipo "Y", a parità di valori nominali. Non è un caso che nel nostro alimentatore i condensatori "X", siano da 330/470nF, mentre gli "Y" siano solo da 4,7nF!

Domanda: come scegliere i valori capacitivi per il filtro di ingresso?
Se avete risposto: usiamo i più elevati valori possibile, avete **quasi** indovinato...
Infatti è evidente che in un filtro L-C, più sono alti i valori di induttanza e di capacità, più la frequenza di filtraggio è bassa, e quindi meno disturbi ad alta frequenza entrerano/usciranno nell'/dall'alimentatore.
Per i condensatori, a parte le considerazioni su costo ed ingombri che aumentano all'aumentare del valore capacitivo, ci sono purtroppo anche altri vincoli derivanti dalle norme di sicurezza. Senza scendere nel dettaglio di quest'ultime, possiamo dire che:

- **Le resistenze R1 e R2, dette resistenze "bleeder", hanno il compito di scaricare velocemente i condensatori del filtro di ingresso, quando si toglie la spina dalla rete.** La costante di tempo dello "scaricamento" è pari al prodotto tra R totale e C totale. La resistenza totale è data dal parallelo tra R1 e R2, e vale quindi R = 410kΩ (trascurando la resistenza serie di L0 e L1).
  La capacità totale è pari alla somma di CX0, CX1, CX2 (trascurando CY1 e CY2 perchè di valore molto inferiore) e quindi C = 1,13μF. Risulta una costante di tempo pari a R * C = 0,46 secondi, cioè **sui poli nudi della spina di alimentazione, appena staccata dalla presa, resterà una tensione pericolosa per un tempo non trascurabile. Se aumenta la capacità, le resistenze di bleeder dovranno essere di valore più basso (per mantenere la costante di tempo invariata), quindi dissiperanno di più e saranno più ingombranti e costose.**
  Notate infine che le stesse resistenze causano uno spreco di energia sempre presente, anche quando l'alimentatore è completamente spento (ma il cavo è collegato). Nel caso in esame questa perdita si traduce in:
  $P_{DISS,R1//R2} = VAC^2 / R = 230^2 / 410000 = 0,13W$
- **I condensatori CY1 e CY2, essendo sottoposti ad una tensione alternata, fanno sempre**

Classe	X1	X2	Y1	Y2
Resistenza a sovratensioni di	4000V	2500V	8000V	5000V

Tabella 1 Classi di condensatori X e Y

scorrere una corrente (anch'essa alternata), verso la terra, che è direttamente proporzionale al valore di capacità. Il calcolo del valore di corrente è molto semplice e ve lo lascio come esercizio (ricordate che l'impedenza di un condensatore a frequenza f è pari a 1 / (2 * π * f * C), ed applicando la legge di Ohm..). Per questo motivo, **non si può esagerare con il valore di capacità dei condensatori "Y"**, in quanto le diverse norme internazionali stabiliscono limiti massimi ben precisi per la massima corrente di dispersione verso terra. Anzi, ci sono alcuni alimentatori, cosiddetti a **"norme medicali"**, dove questi limiti sono ulteriormenti abbassati, quindi sono utilizzabili solo piccoli valori per i condensatori "Y", e di conseguenza si avrà un minore filtraggio dei disturbi.

LE INDUTTANZE DI FILTRO

Per concludere l'analisi del filtro di ingresso, consideriamo le induttanze, cominciando dalla L1. Anche questa induttanza è soggetta a due vincoli contrastanti: avere il valore più elevato possibile per migliorare il filtraggio e nello stesso tempo, avere resistenza serie più bassa possibile per non pregiudicare l'efficienza. Notate infatti che essa è in serie alla tensione di rete, ed è quindi attraversata da una **grande corrente a bassa frequenza (50-60Hz).**

Per ottenere contemporaneamente una bassa resistenza ed un alto valore di induttanza, occorre utilizzare un nucleo con un altissimo valore di A_L, in modo da avvolgere poche spire di filo (di grande sezione).

Tuttavia sapete dalle puntate precedenti che questo implica una saturazione molto precoce del nucleo magnetico...come fare? La soluzione è semplice e geniale: realizzare **due avvolgimenti identici sullo stesso nucleo**, attraversati dalla stessa corrente. Dalla figura 5, la corrente che scorre nel ramo superiore (indicata **ad un**

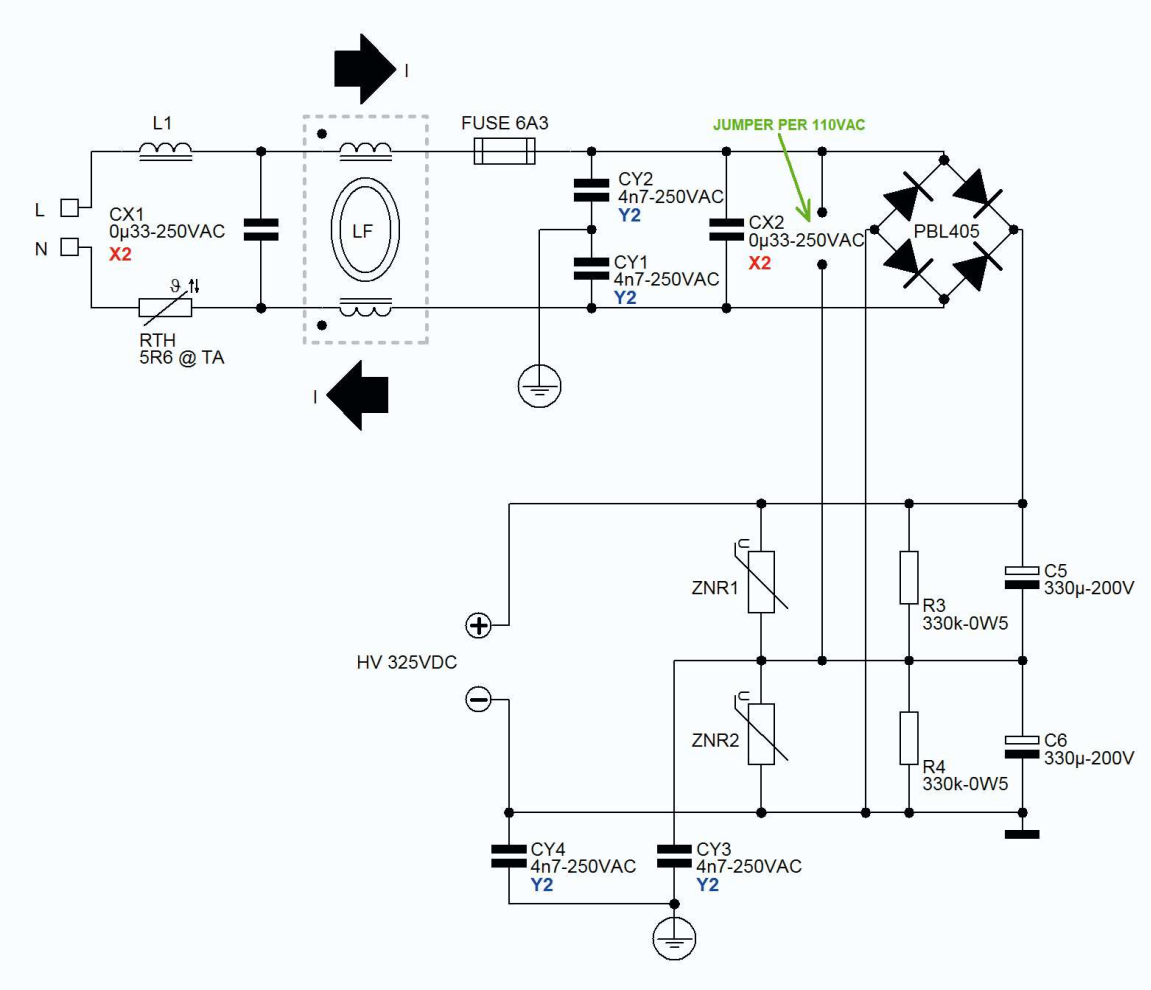


Figura 6 Schema elettrico del convertitore in tensione continua

ITF: INFOMOBILITY FORUM
TELEMATICS 2005

NOVITÀ 2005

IN EVIDENZA

terza edizione

IV Edizione

24 Novembre 2005

CENTRO CONGRESSI LINGOTTO

TORINO

LE AREE TEMATICHE

Organizzato dal

certo istante[1] dalla freccia superiore) **entra** nel "pallino", mentre la corrente nel ramo inferiore esce dal "pallino".
Siccome i due avvolgimenti sono identici, il flusso magnetico generato dalla corrente superiore viene annullato perfettamente da quello generato dalla corrente inferiore.
In pratica, è come se L1 avesse induttanza "zero" per i disturbi differenziali (se entra il disturbo solo sul filo L, ne esce sul filo N e l'effetto di filitraggio è nullo), mentre ha induttanza elevata per i disturbi di modo comune, cioè quelli che si manifestano contemporaneamente su entrambi i fili L e N, rispetto alla terra.
Valori tipici per L1 vanno da 1 a 10mH o più.
A titolo informativo, nella letteratura e nei datasheet troverete indicate le induttanze L1, come *"current-compensated chokes"* oppure *"common-mode chokes/inductors"*, con ovvio significato dei termini.
Dopo questa chiacchierata, diventa immediato capire il motivo della presenza di **L0: il suo scopo è quello di filtrare i disturbi differenziali, ignorati da L1.**

[1] *Infatti la corrente è alternata, e la sua direzione cambia in continuazione.*

CONVERSIONE IN TENSIONE CONTINUA

Dopo il filtro di ingresso, un interruttore a bilanciere permette o meno il passaggio della tensione alternata al circuito stampato principale. Notate come l'interruttore agisca su **entrambi** i fili di rete L e N.
In figura 6 è riportato lo schema completo.

Figura 7 *I condensatori d'ingresso. Sottodimensionati e non LOW-ESR*

Notate che il filtro di ingresso non è ancora terminato! Infatti riconoscerete sicuramente in L1, LF, CX1, CX2, CY1 e CY2[2] **un ulteriore filtro**, in serie al precedente. Tuttavia ci sono alcune differenze:

- É presente una resistenza NTC, indicata come RTH, del valore di 5,6_ a temperatura ambiente.
- É presente un fusibile di protezione.

La resistenza NTC, Negative Temperature Coefficient, ha il compito di limitare la corrente di spunto (*in-rush current* in inglese), che si ha nell'istante esatto di accensione. Infatti la tensione di rete può essere al suo massimo (se si è sfortunati!), che vale 230 * 20,5 = 325V di picco, e si trova come carico il circuito serie costituito dalle varie resistenze serie delle induttanze di filtro (piccole), le resistenze dei diodi del ponte (piccole), ed i condensatori C5 e C6 completamente scarichi. La corrente istantanea che ne risulta, se non ci fosse l'NTC, può tranquillamente superare i 200A!
L'NTC da 5,6Ω, invece, riduce questa corrente di spunto ad un massimo di circa 50A, valore comunque elevato ma sopportabile, poiché dura un brevissimo tempo.
Questa corrente provoca un riscaldamente dell'NTC, che quindi abbassa il suo valore resistivo, e ne diminuisce le perdite durante il normale funzionamento.
Da questo appena detto, deriva un'interessante conseguenza pratica: **una volta spento un alimentatore switching, è buona norma attendere qualche secondo prima di riaccenderlo.**
Questo perchè si deve dare tempo all'NTC di raffreddarsi e ri-aumentare la sua resistenza, in caso contrario all'accensione successiva, la sua resistenza può essere ancora molto bassa e non limitare la corrente di spunto, con conseguenze anche disastrose per l'alimentatore[3].
Dopo il secondo filtro, la tensione di rete arriva finalmente al ponte raddrizzatore, un normalissimo PBL405 da 4A-600V, seguito dai condensatori di livellamento C5 e C6.
Riguardo a questi ultimi, notate il loro utilizzo in serie che permette il raddoppio della tensione di lavoro a scapito del dimezzamento della capacità totale: in pratica sono equivalenti ad un solo condensatore da 165µF-400V, abbastanza sottodimensionato per un alimen-

tatore di questa portata...Come se non bastasse, non sono neanche del tipo LOW-ESR (vedi figura 7): aspettiamoci quindi bassa efficienza e bassa affidabilità.

Le resistenze R3 e R4 hanno il compito di compensare in qualche misura le differenze di capacità tra C5 e C6, che altrimenti potrebbero non suddividersi equamente la tensione continua a cui sono sottoposti.
Ad esempio, se C5 avesse il 25% di tolleranza verso l'alto (330 * 1,25 = **412,5µF**) e C6 il 25% verso il basso (330 * 0,75 = **247,5µF**), su C5 sarebbe presente una tensione di 325 / 660 * 247,5 = 122V mentre su C6 di ben 325 / 600 * 412,5 = 203V, fuori specifica!
Le resistenze R3 e R4 hanno anche la funzione di "bleeder", cioè scaricano i condensatori allo spegnimento dell'alimentatore. Vediamo quanto vale la costante di tempo: R * C = 330000 * 330e-6 = 108,9 secondi! Cioè ci vogliono quasi due minuti per far sì che i condensatori si scarichino a circa 0,37 volte la tensione iniziale, cioè a 120V, ancora pericolosi!
Ovviamente questo vale nel caso non ci siano altri carichi che aiutino a scaricare i condensatori più velocemente. Per stare comunque sempre dalla parte della ragione, mi permetto di ribadire:

Attenzione

In caso di riparazioni su alimentatori switching da rete, PRIMA DI INTERVENIRE, e ad apparecchio spento, misurate sempre con un tester la tensione sui condensatori di ingresso.

Continuando l'analisi dello schema di figura 6, notate i due varistori ZNR1 e ZNR2, che hanno il compito di proteggere i circuiti a valle che vedremo nella prossima puntata, da eventuali, residui, picchi di sovratensione.

[2] *Non sono gli stessi componenti di figura 5 in quanto sono montati su un altro circuito stampato. Le denominazioni sono quelle del costruttore.*

[3] *Esistono naturalmente anche limitatori di corrente attivi, che non soffrono del problema dell'NTC. Tuttavia, in mancanza di informazioni precise sulla particolare apparecchiatura, consiglio di adottare sempre questa precauzione.*

Per concludere, avrete sicuramente notato il ponticello indicato come JUMPER PER 110VAC: è questo il ponticello, non montato nell'alimentatore in oggetto, che permette il funzionamento dell'alimentatore anche in paesi come gli Stati Uniti. Per motivi di risparmio, **l'adattamento per i 110VAC viene effettuato brutalmente convertendo il ponte raddrizzatore a doppia semionda, in un doppio rettificatore a singola semionda, con annesso duplicatore di tensione.**
Vediamo come. Inserendo il ponticello, i due diodi superiori del ponte non vengono più utilizzati (infatti sono in parallelo ai condensatori C5 e C6, ma attraverso loro non potrà mai scorrere corrente). I due diodi inferiori funzionano alternativamente, caricando una volta C5 e l'altra C6. Poiché C5 e C6 sono in serie, si ottiene un tensione continua circa doppia, e quasi identica al caso di funzionamento a 230VAC.
In questo modo tutta la circuiteria a valle vedrà sempre la stessa tensione di alimentazione, circa pari a 325VDC.
In figura 8 è riportato, evidenziato in rosso, lo stadio appena descritto. Notate l'NTC ben sollevata dal circuito stampato per facilitarne il raffreddamento.

CONCLUSIONI

Nella prossima puntata completeremo l'analisi del nostro alimentatore, con lo stadio di potenza, quello di uscita e alcune considerazioni sul PFC. Spero di non avervi tediato, vi ringrazio dell'attenzione e a rileggerci al prossimo mese.

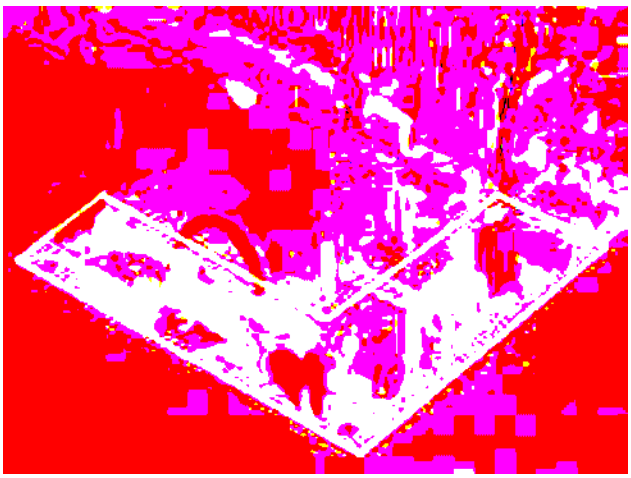


Figura 8 Lo stadio di conversione (senza C5 e C6)

Prima parte
n° 244 - Ottobre 2005
Editor di ISIS - Piazzamento e collegamento dei componenti
Seconda parte
n° 245 - Novembre 2005
ISIS - Creazione dei componenti ed editing avanzato
Terza parte
n° 246 - Dicembre 2005
ARES - Tecniche di base per il piazzamento e lo sbroglio di PCB

Usare PROTEUS:

Nel numero precedente è stato illustrato come creare uno schema elettrico utilizzando le librerie dei componenti incluse nel pacchetto Proteus. Ma che fare se il componente non esiste? La soluzione a questo problema è illustrata nell'articolo di questo mese e, oltre a descrivere come associare il nuovo componente al relativo package, verranno illustrate anche le tecniche di esportazione e stampa.

CREAZIONE DI NUOVI COMPONENTI

In ISIS i nuovi componenti sono creati direttamente insieme al disegno e non esiste un ambiente separato per la creazione o l'editing di dispositivi. Il nuovo componente viene creato piazzando una serie di elementi grafici 2D e i pin, numerando i pin stessi, evidenziando il tutto e, alla fine, richiamando il comando *Crea Dispositivo*.

Troverete molto utile poter creare nuovi componenti nello stesso ambiente, avendo la possibilità di verificare esattamente come il componente apparirà, stabilendo con buona approssimazione quanto grande esso necessita di essere, quanti e quali pin devono essere piazzati su un lato, quali sull'altro e così via. Come esercizio, andremo a costruire un componente, il 7110, il cui aspetto sarà quello di figura 1.

Tale componente è utilizzato nel file *isistut.dsn* che trovate nella cartella *Samples/tutorial*.

Disegno di un Nuovo Componente

La prima cosa da fare è localizzare una zona libera del disegno dove il nuovo componente sarà creato - per posizionare l'area di *Editing* su un'area di disegno libera cliccate con il pulsante sinistro del mouse nella sezione in basso a destra nella finestra di Anteprima.

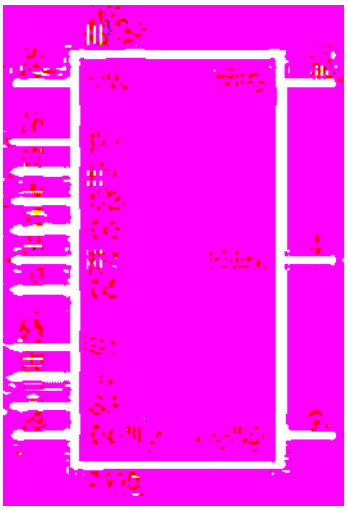


Figura 1 7110 Digital Attenuator finito

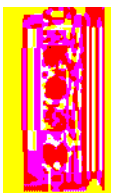


Figura 2 Box Icon

Iniziate disegnando il contorno del nuovo componente. Selezionate l'icona *Box* (figura 2). Vedrete che l'*Object Selector* a sinistra visualizza una lista di *Stili grafici* (figura 3). Uno stile grafico determina le proprietà dell'elemento grafico da disegnare in termini di colore, spessore, tipo di riempimento, colore di riempimento, ecc. Ciascuno stile è un differente insieme di attributi e definisce come le differenti parti appaiono nello schema.

Dal momento che state disegnando i contorni di un componente, selezionate lo stile grafico *COMPONENT*. Posizionate, poi, il puntatore mouse nell'area di *Editing*, premete e tenete premuto il pulsante sinistro del mouse e trascinate un rettangolo. Non preoccupatevi in questa fase della grandezza, potrete sempre cambiarla successivamente. Noterete che avendo scelto uno stile grafico *COMPONENT*, il rettangolo appare con gli stessi attributi di colore, riempimento, ecc. degli altri

ISIS: creazione dei componenti ed editing avanzato

di Maurizio Del Corso
m.delcorso@farelettronica.com

Figura 3 L'Object Selector visualizza gli Stili Grafici

Figura 4 Object Selector e Pin disponibili

componenti sul disegno.
La prossima operazione sarà quella di piazzare i pin sul nuovo componente. Selezionate quindi l'icona *Pin di Dispositivo*. L'*Object Selector* elenca i tipi disponibili di pin (potete anche creare vostri pin personalizzati). Selezionate il pin *Default* (figura 4); la finestra di *Anteprima* fornisce un'anteprima del pin con il nome NAME ed un numero, 99, mentre la base del pin è indicata con un marcatore *Origine* e la parte terminale con un "crocicchio" (il "crocicchio" rappresenta la parte terminale su cui sarà cablato il collegamento) come mostrato in figura 4.

Usate le icone *Ruota* e *Specchia* per orientare l'anteprima affinché la base del pin sia posta a destra e quindi cliccate con il pulsante sinistro del mouse sul lato a sinistra del rettangolo in modo che la base del pin si posizioni esattamente dove volete essa appaia. Piazzate i pin per VIN, D0÷D5, S1÷S3 e DGND. Potete anche usare il tasto della tastiera DOWN per muovere il puntatore di un'unità di griglia e anche il tasto ENTER come sostituto del tasto sinistro del mouse (spesso è più veloce usare questi tasti invece del mouse). Ora cliccate sull'icona *Specchia* e quindi piazzate i pin sulla destra: RFB, IOUT e AGND. Per finire, piazzate due pin, uno in alto e l'altro in basso in corrispondenza dei contorni del rettangolo, cambiandone l'angolo di rotazione con le icone *Ruota* e *Specchia* prima del piazzamento stesso in modo che essi puntino al di fuori del rettangolo del componente; questi pin saranno quelli di alimentazione VDD e VBB ed eventualmente saranno nascosti (questo è il motivo per cui non sono visibili sullo schema di riferimento).

Figura 5 Anteprima del pin

A questo punto, potete riposizionare i pin oppure ridimensionare il rettangolo in modo opportuno. Per spostare un pin, evidenziatelo con il pulsante destro del mouse e quindi trascinatelo nella nuova posizione tenendo premuto il pulsante sinistro del mouse; per cambiarne l'orientamento, usare le icone *Ruota* e *Specchia*. Per ridimensionare il rettangolo, evidenziatelo prima premendo il pulsante destro del mouse e, quindi, cliccando e tenendo premuto il pulsante sinistro su una delle otto maniglie di trascinamento (quei piccoli riquadri neri posti ai vertici e sulle mediane dei lati del rettangolo). Trascinate la maniglia fino alla nuova posizione. Se avete cambiato la larghezza del componente, dovrete spostare anche i pin. Evidenziateli tutti trascinando un rettangolo attorno ad essi (con il pulsante destro del mouse) in modo che usando il comando *Sposta* questo li riposizioni tutti. Quando avrete ultimato di ridimensionare il rettangolo e posizionato i pin dovrete annotarli con il loro nome e numero e dovrete assegnare loro una tipologia. La tipologia del pin (input, power, pull-up, ecc.) è usata dallo strumento *Electrical Rules Check* allo scopo di garantire che solo pin della corretta tipologia siano collegati tra loro.
Assegneremo prima i nomi, la tipologia e la visibilità dei pin. Dobbiamo evidenziare ciascun pin con il pulsante destro e quindi editarlo cliccando il pulsante sinistro del mouse; il pin

visualizza la finestra di dialogo *Modifica Pin*.
Editare ciascun pin di seguito come indicato:

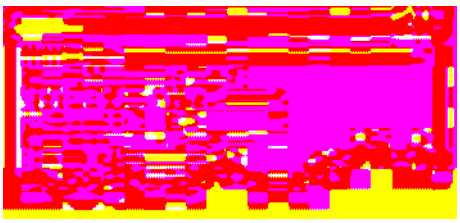


Figura 6 *Finestra di dialogo Crea Dispositivo*

- Digitate il nome del pin nel campo *Nome*. Lasciate vuoto il campo *Number* in quanto lo assegneremo poi con lo strumento *Strumento di assegnazione proprietà*.
- Scegliete una tipologia opportuna per il pin - *Output* per il pin IOUT, *Power* per VDD, VBB, AGND e DGND, e *Input* per i rimanenti.
- Indicate i pin che devono essere nascosti deselezionando l'opzione *Draw body* - i pin VDD e VBB sono entrambi pin standard di alimentazione e pertanto possono essere nascosti. I pin AGND e DGND non sono standard e quindi richiedono di essere visibili in modo che possano essere collegati in modo opportuno agli altri componenti nel disegno che li usano.
- Premete il pulsante OK quando terminato.

Ciò che abbiamo fatto sinora è quello di mettere insieme una serie di elementi grafici che costituiscono le parti del componente desiderato. La fase finale sarà quella di usare il comando *Crea Dispositivo* che le trasformerà nella parte di libreria ISIS.

Salvare il componente in libreria

Evidenziate tutti i pin e i contorni del componente - il metodo più semplice è trascinare con il pulsante destro del mouse un'area attorno le parti da evidenziare. Selezionate ora il comando *Crea Dispositivo* dal Menu Libreria; dovreste vedere la finestra di dialogo *Crea Dispositivo* (figura 6). Digitate nel campo *Name* il nome `7110` e la lettera `U` nel campo *Prefisso*, come indicato in figura 6.

Selezionate i pulsanti *Succ.>* quindi *Aggiungi/Modifica* per aggiungere il tipo di contenitore (*packaging*).
Selezionate il pulsante *Aggiungi* in alto della finestra di dialogo. Nel nostro caso vogliamo aggiungere un DIL16 dalla libreria PACKAGE. Trovata l'impronta (footprint) nell'elenco, per aggiungerla al componente da salvare cliccateci due volte sopra. La finestra di dialogo dovrebbe ora apparire come in figura 7.
La fase successiva è quella di numerare i terminali dei pin. Potete sempre aggiungere la numerazione manualmente nella colonna "A", ma spesso è più facile (e più sicuro) farlo in modo interattivo. Cliccate con il mouse nelle colonna "A" a lato del terminale "AGND", come mostrato in figura 7.

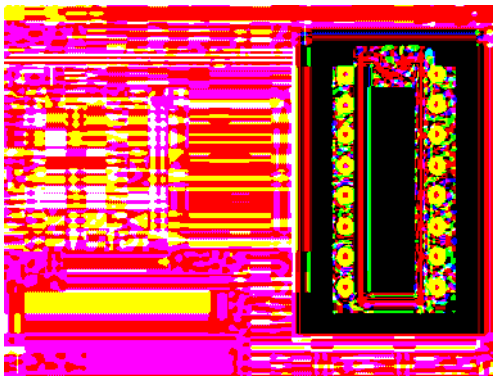


Figura 7 *Strumento Packaging con DIL16 pronto per l'assegnazione dei pins*

Il pin "AGND" è il numero 2 - cliccate sulla piazzola (pad) "2" sull'anteprima dell'impronta come mostrato in figura 8. Dovreste vedere la piazzola evidenziarsi e il numero del terminale "2" sarà assegnato a "AGND" mentre il cursore si sposterà automaticamente sul terminale "D0".
Potete cliccare con il mouse sulle opportune piazzole (8 per

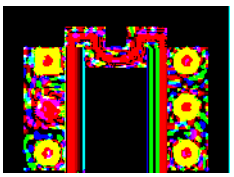


Figura 8 *Assegnazione interattiva tramite lo strumento Packaging*

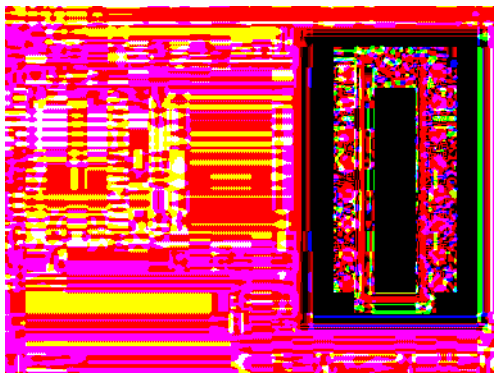
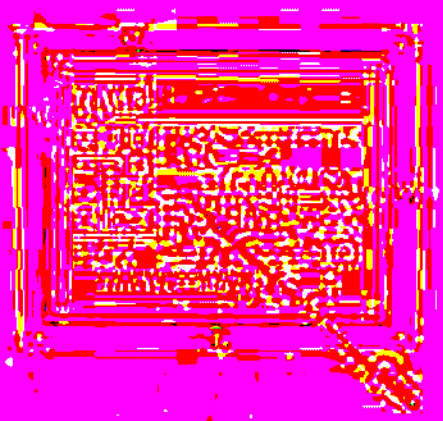


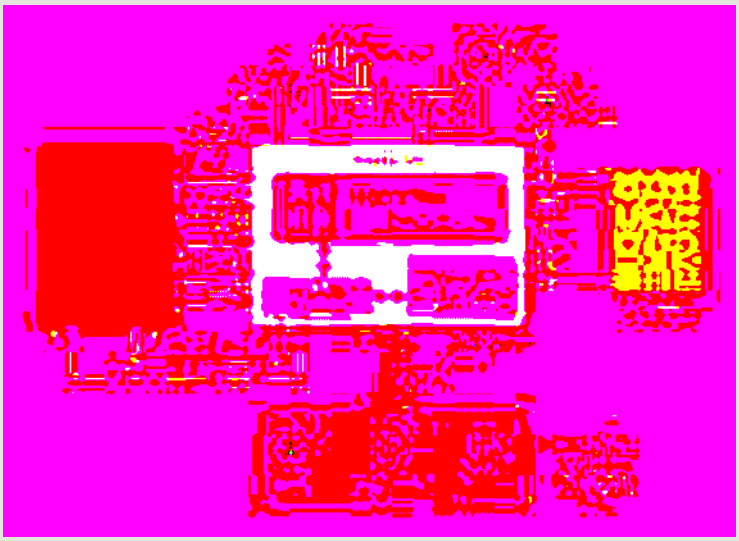
Figura 9 *Strumento Packaging dopo la numerazione dei pin*

Un innovativo LCD controller:



Gli iLCD sono una famiglia di innovativi display grafici nati per abbattere drasticamente i tempi di sviluppo.
È possibile creare in maniera molto semplice e rapidissima numerosissime schermate utilizzando tutti i fonts di windows, animazioni, templates e macro, grazie ad un software gratuito.

Solo con questi prodotti è possibile lasciare al display tutta la gestione della grafica.
Sono disponibili versioni con touch screen.



CARATTERISTICHE HARDWARE

- Supporto per tastiera fino a 32 tasti
- Supporto per 6 LED (on/off/blinking)
- 4 ingressi analogici o digitali
- Cotrollo per 2 relè (max 100mA)
- 198Kbyte FLASH 512Byte EEPROM
- Possibilità di gestire un alimentatore ATX
- Contrasto e retroilluminazione gestibili via software
- Firmware aggiornabile via RS232 o USB

INTERFACCE

- USB
- RS232 (110Baud.. 115200Baud)
- RS422 e RS485
- I2C Bus

netwaves

INWARE SRL
Via Cadorna, 27/31
20032, Cormano, MI - Italy

Tel.: +39 02.66504794 - 02.66504755
Fax: +39 02.66508225

info@inware.it
www.netwaves.it

D0, 7 per D1, eccetera) sull'anteprima dell'impronta del contenitore, prendendo a riferimento il componente 7110 visto precedentemente. Ciò assegnerà la numerazione dei pin al componente - quando avrete terminato, il tutto dovrebbe apparire come mostrato in figura 9.
Ora tutte le piazzole dovrebbero essere evidenziate (a meno di errori) e non vi resta che cliccare il pulsante *Assegna Package* e quindi premere sul pulsante OK per ritornare alla maschera principale di *Crea Dispositivo*.
Dovreste ancora una volta vedere l'anteprima del contenitore scelto.
Premete ora sul pulsante *Succ.>* fino a che viene visualizzata la maschera *Selezione Libreria*, mostrata in figura 10.

A questo punto, specificate una categoria e una sottocategoria per la parte creata. Suggeriamo "Analog ICs" e "Miscellaneous", come mostrato sopra. Potete anche digitare una descrizione della parte e scegliere la libreria dove questa sarà salvata, anche se tipicamente l'unica libreria non protetta da scrittura è quella utente (USERDVC).

Un tocco finale...

Ora che il componente 7110 è stato definito, potete piazzarlo e collegarlo normalmente.
Potete ora annotare le parti rimanenti della sezione circuitale appena disegnata - per annotare le nuove parti nello schema senza disturbare quelle esistenti usate *Annotazione Automatica* in modalità incrementale.
Dovreste ora piazzare il bordo del foglio ed il cartiglio del disegno. Selezionate quindi l'icona *Box*, riducete il disegno fintanto che vedete l'intero contorno del foglio (blu scuro) e disegnateci sopra il bordo. Questa operazione deve essere fatta in quanto il contorno del foglio in blu scuro serve solo da riferimento e non viene stampato su carta. Se volete la stampa del bordo dovete piazzarne uno nel disegno.

Figura 10 *Maschera di assegnazione della categoria e della libreria*

Il cartiglio (header block) merita di essere discusso in maggior dettaglio. Esso è, in effetti, simile a qualsiasi altro simbolo che potreste usare, ad esempio, per il logo della vostra azienda. Viene comunque fornito un cartiglio di default HEADER, ma potreste ridefinirlo per i vostri scopi.
Per piazzare il cartiglio, selezionate l'icona *Simbolo* e quindi cliccate sul pulsate P del *Object Selector* per visualizzare la finestra di dialogo *Selezione Libreria Simboli*.
Prelevare simboli dalla libreria è simile all'operazione che avete già effettuato quando avete prelevato una parte dalla libreria dei componenti, eccetto che non esiste alcun sistema di categorie per i simboli. Selezionate l'oggetto HEADER dalla libreria SYSTEM e chiudete la finestra di dialogo. Puntate ora nella zona in basso del disegno dove desiderate porre il cartiglio, premete sul pulsate sinistro del mouse e trascinatelo in posizione.
Alcuni campi del cartiglio sono riempiti automaticamente; altri come Design Title, Sheet Title, Author e Revision richiedono che l'utente digiti altre informazioni usando i comandi *Modifica Proprietà Progetto* e *Modifica Proprietà Foglio* nel Menu *Progetto*.
Notate che il campo *Nome Pagina* nella finestra di dialogo *Modifica Proprietà Foglio* si differenzia dal campo *Titolo Pagina* - *Nome Pagina* è un breve identificatore che etichetta ciascuna pagina di un progetto gerarchico.
Il campo *Titolo Pagina* è una descrizione completa del circuito di quel foglio ed è questa ad apparire nel cartiglio.
Ingrandite il foglio per vedere gli effetti delle modifiche.

SALVATAGGIO, STAMPA E PLOTTING

Potete salvare il vostro lavoro in qualsiasi momento tramite il comando *Salva* del Menu *File*, (non potete se state usando la versione Demo, mentre invece è consentito nella versione Lite) ed ora è arrivato il momento di farlo! L'opzione *Salva con nome* permette di

Figura 11 *Finestra di dialogo Stampa configurata in modo da centrare il disegno sulla pagina*

salvare il file con un nome differente da quello originale.
Per stampare lo schema, selezionate all'inizio il dispositivo di stampa corretto usando il comando *Imposta Stampante* nel Menu *File*. Ciò attiva la finestra di dialogo standard di Windows per la scelta e la configurazione dei dispositivi di stampa disponibili. I dettagli sono quindi dipendenti dalla particolare versione di Windows e dalla vostra stampante - per tutti i dettagli consultate la documentazione di Windows e della vostra stampante. Una volta selezionata la stampante, chiudere la finestra di dialogo e, per stampare il vostro schema, selezionate l'opzione *Stampa* dal Menu *File*. Vi sono parecchie opzioni in questa finestra di dialogo, ma per i nostri scopi centreremo solo lo schema sulla pagina ed infine la stamperemo. Cliccate sull'anteprima di stampa *Anteprima* di stampa e selezionate l'opzione *Allinea* uscita al centro così come mostrato nella figura 11.
Se state usando la versione demo, potete stampare uno dei disegni forniti non modificati. Usate il comando *Apri* nel menu *File* per caricare tali disegni tra quelli già forniti.

ALTRE INFORMAZIONI SULLA CREAZIONE DI DISPOSITIVI

Come costruire una parte a Sezioni Multiple

Definiremo ora un'opportuna libreria per la parte TL074 quadruplo op-amp. Questa parte è costituita da quattro sezioni separate in un singolo contenitore, pertanto vi insegneremo come creare componenti a sezione multipla utilizzando lo *Strumento Packaging*.

La figura 12 mostra il nuovo op-amp prima che questo sia inserito nella libreria. L'op-amp è costituito da elementi grafici 2D, 5 pin e un marcatore origine. Vedremo due modi per costruire questa parte. L'approccio più semplice è utilizzare il simbolo predefinito OPAMP. Procedere quindi in questo modo:

Figura 12 *Un elemento del TL074 Quadruplo op-amp prima della sua costruzione*

- Cliccare sull'icona *Simbolo* quindi cliccare nuovamente sul pulsante P nell'angolo a sinistra del *Symbols Object Selector*. Questa operazione farà apparire la finestra di dialogo *Selezione Librerie*.
- Cliccate due volte su OPAMP nella libreria *System* e chiudete la finestra di dialogo usando il pulsante "Riduci a Icona" di Windows sulla destra della barra Titolo.
- Posizionate il mouse in una zona libera dell'area di *Editing* e usate il pulsante sinistro del mouse per piazzare l'op-amp. Il componente apparirà con lo stile grafico *COMPONENT* in quanto è stato creato in questo modo.

Ora dovete piazzare i pin attorno il simbolo del componente. Tale procedimento è lo stesso che avete già usato in precedenza per creare l'attenuatore 7110:

- Selezionate *Pin di dispositivo* per ottenere la lista dei pin disponibili e selezionate il tipo di *Default*.
- Usate le icone *Ruota* e *Specchia* per orientare ciascun pin prima di piazzarlo sul disegno.
- Quando tutti i pin sono correttamente posizionati, editate ciascun pin evidenziandolo in sequenza, prima con il pulsante destro del mouse e quindi cliccandovi sopra con il pulsante sinistro. Usate la finestra di dialogo *Modifica Pin* per annotarlo con la corretta tipologia e nome. Il nome è necessario in quanto faremo riferimento a questo quando useremo lo *Strumento Packaging*, tuttavia scegliamo di non visualizzare il nome del

pin sul componente (l'uso dei pin è implicito per l'op-amp scelto). Quindi assicuriamoci che l'opzione *Mostra Nome* non sia selezionata. Non vi è alcuna necessità di digitare la numerazione dei pin, in quanto la definiremo usando lo *Strumento Packaging*.

I pin di alimentazione sono denominati V+ e V- ed hanno la tipologia elettrica *Power*.
Se cercate di piazzare tali pin sui lati obliqui del simbolo dell'op-amp, vi accorgerete che il terminale non è in linea con la griglia e, viceversa se cercate di allineare alla griglia il terminale del pin (marcato con un "crocicchio") la base del pin non tocca il lato obliquo. In tal caso, allineate il terminale del pin alla griglia ed estendetene la base fino al simbolo dell'op-amp usando una linea con l'icona *Linea Grafica 2D* ed aumentando la risoluzione di snap. I pin d'ingresso sono denominati `+IP` e `-IP` e la tipologia corretta è *Input*. Il pin di output ha per nome `OP`, mentre la sua tipologia è *Output*.
La fase finale sarà quella di posizionare il marcatore *Origine*.
Selezionare l'icona *Marcatore* per visualizzare in *Object Selector* un elenco disponibile di simboli. Selezionate il marcatore *Origin* e piazzatelo al centro del simbolo grafico dell'op-amp.
Il marcatore *Origin* è visualizzato come un piccolo rettangolo con una croce ed indica ad ISIS il riferimento del componente rispetto al puntatore del mouse quando la parte sarà trascinata o posizionata nel disegno.
Abbiamo completato la marcatura del componente. Evidenziate tutti i vari elementi grafici - il simbolo dell'op-amp, i pin ed il marcatore *Origin* - trascinando un rettangolo di selezione con il pulsante destro del mouse, e quindi usate il comando *Crea Dispositivo* nel Menu *Libreria*.
Procedete come segue:

- Digitate il nome nel campo *Nome dispositivo*, es. TL074 e il *prefisso* "U"
- Cliccate sul pulsante *Succ.>* per visualizzare la pagina *Packaging* e cliccate *Aggiungi/Modifica* per aprire lo strumento packaging.

Usare lo strumento packaging

Lo strumento *Packaging* fornisce l'ambiente grafico in cui assegnare ad una parte shematica uno o più impronte PCB.
Viene creata una tabella di pin numerata e denominata in modo differente per ciascuna impronta PCB in quanto ogni contenitore può avere un differente numero di pin, anche per la stessa parte schematica.
Con lo strumento di Packaging aperto creiamo l'impronta da assegnare al componente:

- Cliccare il pulsante *Aggiungi*. Ciò aprirà l'elenco della libreria di ARES.
- Selezionate la libreria PACKAGE, e cliccate due volte sulla parte DIL14.

Quindi effettuate le seguenti modifiche:

- Cambiate il numero di elementi da 1 a 4. Ciò perché il componente ha quattro sezioni op-amp all'interno di un contenitore DIL14.
- Definire i pin V+ e V- come pin comuni. Ciò significa che i quattro elementi hanno in comune due pin di alimentazione che voi potrete collegare solo ad una sezione o a tutte. Tali collegamenti saranno considerati interconnessi.
- Cliccate l'opzione "*Consento lo scambio tra porte nel layout*". Questa opzione specifica che gli elementi sono identici e che pertanto possono essere scambiati durante lo sbroglio del PCB in ARES.

Dovreste vedere una finestra simile a quella riportata in figura 13.
Assegnate ora la numerazione dei pin. Per aiutarvi, la figura 14 mostra la numerazione dell'op-amps che stiamo costruendo:
Procedete come segue:

- Cliccate sulla colonna "A" per il pin +IP.
- Cliccate sia sulla piazzola "3" dell'anteprima del contenitore oppure digitate "3" con la tastiera e quindi premete TAB. In entrambi i modi, il pin "3" sarà evidenziato sull'anteprima per mostrare che è stato assegnato. Il cursore si sposterà automaticamente sulla riga -IP.
- Ora ripetete il procedimento per gli altri

I CD-ROM di Fare

Annata 2003 e 2004

22 Uscite di Fare Elettonica in 2 CD-ROM!

Due annate complete in formato ACROBAT (PDF).

Potrai sfogliare comodamente tutte le riviste uscite nel 2003 e nel 2004 e stampare gli articoli di tuo interesse.

I CD delle annate sono acquistabili separatamente oppure insieme ad un prezzo scontato.

Vitamina C

Il corso completo Vitamina C in formato ACROBAT (PDF)

Tutte le lezioni del corso base al linguaggio C comodamente consultabili e stampabili in alta risoluzione.

Notevolmente migliorata e completata rispetto alla versione originale pubblicata sulla rivista.

Il **CD-ROM Vitamina C** contiene una sezione "Contenuti Speciali" tutta da scoprire.

PIC® Microcontroller By Example

Il corso completo PIC® Microcontroller By Example in formato ACROBAT (PDF)

Tutte le lezioni pronte per la consultazione con i sorgenti dei progetti immediatamente utilizzabili nelle tue applicazioni.

Il modo migliore per avere sempre sottomano la soluzione per il tuo progetto con i **PICmicro®**.

Il **CD-ROM PIC® Microcontroller By Example** contiene una sezione "Contenuti Speciali" tutta da scoprire.

Novità

Tutto sulle Smartcard

La raccolta completa degli articoli sulle smartcard in formato ACROBAT (PDF)

Gli articoli, i progetti e i download relativi agli articoli sulle Smartcard in un unico CD-ROM pronto da consultare ed utilizzare.

Contiene i progetti completi del lettore di smartcard **UniReader** e del **SimCopier** per duplicare il contenuto della rubrica della vostra Sim card.

Ordinali subito su www.farelettronica.com oppure telefona allo 02.66504794

Inware Edizioni - Via Cadorna, 27/31 - 20032 Cormano (MI)

pin, fino a che tutti sono stati assegnati, cosa che l'anteprima indicherà evidenziando tutte le piazzole corrispondenti. Naturalmente controllate di non aver dimenticato qualche pin.

Infine, cliccate sul pulsate *Assegna Packages* per ritornare alla finestra di dialogo *Crea Dispositivo*, e salvate la parte nella libreria USERDVC, esattamente come avete fatto per il componente 7110.

Come derivare componenti simili

Avendo costruito un TL074 potete quasi istantaneamente definire modelli derivati come TL064 e TL084. Piazzate semplicemente un TL074, evidenziatelo e chiamate il comando *Crea Dispositivo*; cambiate il nome in TL064 (o qualsiasi altro) e salvatelo. Non poteva essere più semplice!
Se avete bisogno di aggiungere altri particolari al TL074 - come ad esempio più simboli grafici, potete semplicemente aggiungerli sopra il TL074 e chiamare il comando *Crea Dispositivo*. In alternativa, se il TL074 era sostanzialmente corretto ma richiedeva di essere ritoccato prima di adattarsi al nuovo componente derivato potete scomporlo nei suoi elementi grafici, evidenziandolo prima e poi selezionando il comando *Scomponi* dal Menu *Libreria*, editando o aggiungendo i particolari richiesti e quindi marcando il nuovo dispositivo.

Sostituzione di componenti in un disegno

Potete ora sostituire i quattro filtri op-amps con le parti appropriate TL074. Per sostituire i componenti con un tipo simile, prelevate il nuovo dispositivo, assicuratevi che il mouse sia dentro il componente da sostituire, cliccate e tenete premuto il pulsante sinistro del mouse ed infine trascinate il nuovo componente in modo da far combaciare i pin del nuovo con quello vecchio. ISIS trasferirà i collegamenti dal vecchio componente a quello nuovo, mantenendo intatte le altre informazioni (es. Il suo riferimento, ecc.).

Simboli e libreria di simboli

Evidenziate le tre linee che formano la squadretta attorno gli ingressi del 7110. Chiamate quindi *Crea Simbolo* dal Menu *Libreria*, digitate TEST come nome del simbolo e premete ENTER. Selezionate ora l'icona *Simbolo*. Vedrete apparire nell'*Object Selector* la parte TEST. Prelevate tale parte, provando a piazzarla sul vostro disegno. L'uso di questi simboli è abbastanza comune come OPAMP, che è comune a molti componenti, ma anche per i logo, emblemi ecc.
Il blocco HEADER usato per il cartiglio fa un uso speciale dei simboli. Il simbolo di default è stato creato da linee 2D, da un rettangolo ed altre speciali primitive di testo che sono automaticamente sostituite dal testo associato ad esse. Ad esempio, un oggetto testo con la stringa:

```
@DTITLE
```

visualizzerà il titolo del disegno una volta che questo è stato digitato nella finestra di dialogo del comando *Modifica Proprietà Progetto*.

GENERAZIONE DI RAPPORTI

Ora che il disegno è stato completato, potete generare i rapporti disponibili come la Netlist, l'elenco dei Materiali o Bill of Materials (BOM) oppure lanciare l'Electrical Rules

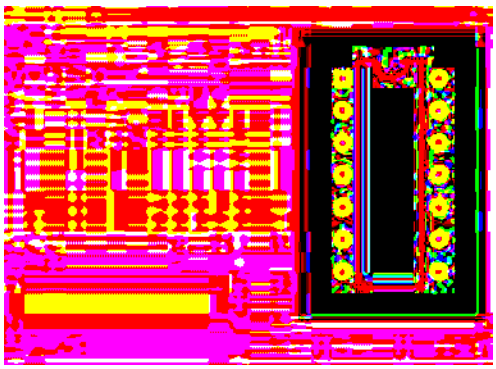


Figura 13 *Lo strumento Packaging pronto per l'assegnazione dei pin*

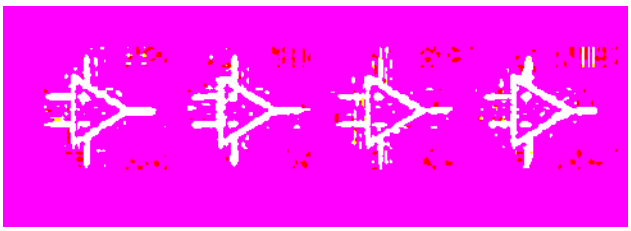


Figura 14 *I quattro elementi del nostro TL074*

Check (ERC). Ciascun rapporto è generato chiamando il comando corrispondente nel Menu *Strumenti*.
I risultati del rapporto sono visualizzati in una finestra di testo dalla quale possono essere salvati in un file selezionando il pulsante "Salva con nome" oppure copiati nella clipboard per l'uso in altri programmi con il pulsante "Copia negli appunti"; il pulsante "Chiudi" permette di chiudere la finestra di testo e di ritornare nell'editor di disegni. Notare che l'ultimo rapporto non viene perduto chiudendo la finestra di testo - per rivederla è sufficiente selezionare ancora il comando *Visualizzatore Testi* nel Menu *Sistema*.
I servizi offerti dal rapporto Bill of Materials dovrebbero essere auto esplicativi, tuttavia è possibile ottenere una serie di servizi ulteriori. Il rapporto Electrical Rules Check conterrà una serie di errori poiché il circuito presentato nel presente tutorial non è un progetto completo.

PROGETTI DI GRANDI DIMENSIONI

In questa ultima sezione prenderemo in esame un progetto preesistente - EPE.DSN. Questo progetto è un emulatore e programmatore di EPROM a microprocessore, ed è gestito con una tecnica schematica detta gerarchia a fogli multipli.
Come tale, esso rappresenta un buon esempio di relativa complessità circuitale che potete ottenere facilmente dal vostro sistema ISIS. Il progetto EPE è strutturato in 3 fogli principali in formato A3 (Processor, Emulator e PSU). Sono usati alcuni sotto-schemi per rappresentare i banchi di Emulazione RAM (4 banchi per 32k di emulazione) e 6 sezioni di alimentazione (PPSU) che sono necessarie per coprire il range della serie EPROM 27.
Caricate in ISIS il progetto usando il comando *Apri* nel Menu *File*, e selezionate EPE.DSN dall'elenco. Troverete il file in questione sotto "Samples\Schematic & PCB Design" nella cartella di installazione del vostro Proteus.
In alternativa, potete visionare i disegni forniti chiamando il file di guida SAMPLES nel Menu *Giuda*.
Il primo foglio è la CPU - dateci uso sguardo usando sia lo scorrimento che le opzioni di zoom.
Per vedere altri fogli usate il comando *Vai alla pagina* nel Menu *Progetto*. Scegliete la seconda voce dall'elenco e dopo un istante sarà visualizzato il foglio Emulator Control. Ridimensionate il foglio in modo da vederlo interamente.
I 4 rettangoli blu sono i sotto-circuiti. Il testo sopra ciascun sotto-circuito è l'identificatore ID (simile al riferimento delle parti) ed il testo in basso è il nome del circuito.
Potete usare lo zoom anche nei sotto-circuiti: ponete il puntatore del mouse su uno dei sotto-circuiti ed usate la sequenza di tasti assegnata (per default è CTRL+C). ISIS passa velocemente dal foglio Emulator Control al foglio banco RAM. Date uno sguardo al circuito, in particolare prendete nota della numerazione dei componenti. Per uscire da un sotto-circuito usate la sequenza assegnata (per default è CTRL+X).
Entrate in un altro sotto-circuito del banco RAM e verificate la numerazione dei componenti con quello precedente - sebbene il circuito sia lo stesso (se modificate qualcosa in un sotto-circuito, la modifica si riflette negli altri) ciascun componente ha una diversa numerazione; questa è la caratteristica dello strumento di *Annotazione globale progetto* al lavoro.
Ora che conoscete come gestire i vari fogli in un progetto gerarchico potete esplorare il resto del progetto EPE. Tale progetto rappresenta un buon esempio di circuiti analogici, digitali e a microprocessore, il che dimostra quanto ISIS sia ben adatto nelle varie tipologie di schemi elettrici.

COS'ALTRO VI ATTENDE

Dal prossimo numero inizieremo ad occuparci di ARES, quindi di come realizzare un circuito stampato. Per il momento vi suggeriamo di fare un po' di pratica con ISIS esplorando anche le altre funzionalità del programma.
Vi ricordiamo che sul sito di Fare Elettronica è disponibile il Forum dedicato a Proteus, utile per scambiare idee e suggerimenti.
L'appuntamento è in edicola il prossimo mese.

Ventiduesima parte
n° 244 - Ottobre 2005
Tecniche di debug

Ventitreesima parte
n° 245 - Novembre 2005
Gestione delle interruzioni

Ventiquattresima parte
n° 246 - Dicembre 2005
Sistemi operativi

Vitamina C:

In questa puntata vedremo cosa sono le interruzioni, quando e come utilizzarle, e come gestirle programmando in C. Saranno presi in considerazione gli aspetti più interessanti e critici legati all'uso delle interruzioni nei nostri programmi, e verrà mostrato come evitare alcuni errori comuni.

INTRODUZIONE

Normalmente l'esecuzione di un programma è strettamente sequenziale, e anche in presenza di salti o chiamate a subroutine è sempre possibile seguirne o prevederne l'evoluzione. In altre parole un qualsiasi programma, soprattutto nel caso di sistemi embedded, è praticamente un loop infinito (il classico while(1){…}), che esegue ciclicamente tutte le operazioni necessarie al controllo ed alla gestione del sistema (figura 1).
In molti casi però questo tipo di funzionamento non risulta adeguato alla gestione di certi problemi. Spesso infatti è richiesto che al verificarsi di certi eventi esterni (più o meno imprevedibili), il programma risponda immediatamente, per poi tornare al suo normale funzionamento. Per ottenere questa caratteristica viene utilizzato il meccanismo delle interruzioni (*interrupt* in inglese), che è di solito messo a disposizione dalla maggior parte di microprocessori e microcontrollori. Una interruzione è quindi un segnale esterno che provoca un temporaneo ed immediato cambiamento del flusso di esecuzione del programma.
Un programma che utilizza le interruzioni ha di solito una struttura diversa, in quanto oltre al loop principale, alcune funzioni sono svolte dalla (o dalle) routine di interruzione (figura 2).
Chiaramente quindi è necessario un progetto molto più accurato, e questo è ancora più importante quando si utilizza un linguaggio ad alto livello (in assembly risulta relativamente più diretto e naturale).

Caratteristiche delle interruzioni

La principale caratteristica delle interruzioni è la loro imprevedibilità: possono essere generate in un qualsiasi momento, e quindi interrompere il programma in un qualsiasi punto. Se le cose sono state progettate bene ci sarà un immediato salto alla routine di servizio delle interruzioni (*Interrupt Service Routine*, o ISR), che svolgerà il compito che le è stato affidato, e terminata questa, verrà ripre-

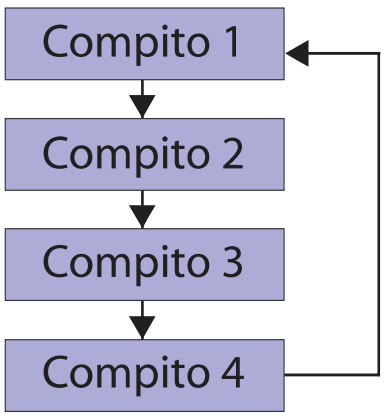


Figura 1 *Struttura di un programma puramente sequenziale*

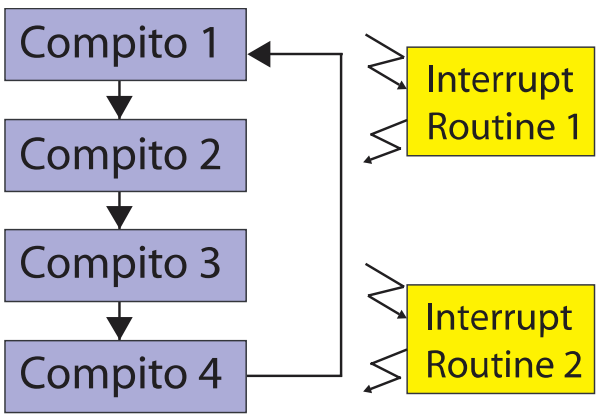


Figura 2 *Struttura di un programma che utilizza le interruzioni*

Gestione delle interruzioni

di Antonio Di Stefano
a.distefano@farelettronica.com

sa la normale esecuzione del programma da dove era stato interrotto, come se non fosse successo niente. Il funzionamento è simile a quello di una qualsiasi subroutine, solo che in questo caso la sua chiamata avviene in maniera imprevedibile in un qualsiasi punto del codice. Alcuni esempi tipici dell'uso di interruzioni sono i seguenti:

1. Segnalazione di eventi esterni improvvisi (allarmi, pressione di pulsanti, superamento di alcune soglie).
2. Sincronizzazione o temporizzazioni (utilizzando l'interrupt di un timer esterno).
3. Sapere quando un dato è disponibile o una periferica è pronta, senza dovere attendere esplicitamente (cioè senza dovere fare un *polling*). Un esempio tipico di questa tecnica è l'interrupt fornito da una UART alla ricezione di un carattere, o la fine si una conversione da parte di un ADC.

Usando questo approccio il microprocessore non avrà bisogno di controllare in continuazione questi parametri, ma riceverà una segnalazione solo quando sarà successo qualcosa che necessita attenzione. È chiaro che se si sfruttando queste tecniche, il codice assume una struttura diversa da quella tipicamente sequenziale a cui siamo abituati.

Meccanismi d'interruzione

Prima di vedere come trattare le interruzioni in C è il caso di fare una breve panoramica sui diversi meccanismi hardware che stanno alla base delle interruzioni, e che come vedremo possono influire molto sulla struttura del software. Nella maggior parte di microprocessori gli interrupt sono comandati dallo stato di alcune linee dedicate (indicate di solito come INT o IRQ), è sufficiente comandare opportunamente queste linee per richiamare la (o le) ISR. Nei microcontrollori invece le linee di interruzione sono spesso associate normali a piedini di I/O, ed il loro funzionamento è programmabile. Non solo, anche le periferiche interne possono generare interruzioni se abilitate (UART, ADC, DMA, timer…). È possibile riconoscere la periferica che ha generato l'interruzione leggendo un apposito registro (del dispositivo o del controllore delle interruzioni, se presente), o utilizzando il "**vettore d'interruzione**" fornito dalla periferica (quando la vettorizzazione è supportata dal sistema).
Se si procede con il polling, o a volte anche usando un controllore, allora la ISR è quasi sempre una sola. Se invece si dispone di un vettore d'interruzione, il processore ricava da questo un indirizzo di memoria, in cui va a cercare la ISR. In questo caso ci saranno tante ISR quante sono le periferiche interrompenti (più o meno).
Gli interrupt possono essere abilitati e disabilitati via software, sia singolarmente che nella loro totalità, solo l'NMI (*Non Maskable Interrupt*, interrupt non mascherabile), quando presente, non può essere disattivato in alcun modo via software.
Indipendentemente da tutte le varianti viste prima, occorre notare una cosa molto importante. Quando si verifica un'interruzione, per potere saltare alla ISR e riprendere poi l'esecuzione come se non fosse accaduto nulla, occorre salvare il "**contesto di esecuzione**". Cioè in primo luogo deve essere salvato il Program Counter (cioè l'indirizzo dell'istruzione che si stava per eseguire), ed eventualmente altri registri sensibili, come ad esempio l'accumulatore ed il registro di stato. La ISR infatti potrebbe modificare il contenuto dei registri, e quindi riprendendo la normale esecuzione del programma, i valori potrebbero differire da quelli precedenti all'interruzione. In genere il Program Counter viene salvato dal processore stesso nello stack, gli altri registri devono essere salvati manualmente all'inizio della ISR, o usando lo stack, oppure cambiando il set di registri utilizzati, quando possibile, e poi ripristinati prima di uscire dalla ISR.

Gestione delle Interruzioni in C

Come visto negli scorsi paragrafi esistono molti

aspetti di "basso livello" che bisogna considerare quando si decide di utilizzare e gestire le interruzioni in C, non è sufficiente limitarsi alla sintassi o allo stile del codice.
L'ANSI C non prevede nessuna funzione nativa o di libreria per la gestione delle interruzioni, bisogna pertanto affidarsi a librerie o estensioni proprietarie fornite dai vari compilatori. Non solo, queste estensioni possono differire molto da un prodotto ad un altro, data anche la grande varietà di meccanismi offerti dalle varie architetture target.
Nella maggior parte dei casi il compilatore permette di utilizzare delle keyword o dei nomi appositi per designare una normale funzione come ISR. In molti casi ad esempio è sufficiente chiamare la funzione ISR con un nome riservato, oppure aggiungere la parola "interrupt" o "int_hadler" prima o dopo la dichiarazione della funzione, ad esempio:

```
/* MikroC per PIC */
void interrupt(void);
/* GCC (alcune versioni) */
void int_UART(void) __attribute__
((interrupt_handler));
```

Nei casi in cui il processore supporti interruzioni vettorizzate, si possono specificare più ISR, e per ciascuna è necessario aggiungere anche qualche informazione sul vettore, ad esempio:

```
/* ZDS-II per Z8 */
void interrupt mia_ISR(void)
{
  SET_VECTOR(INT0,int0_handler);
  …
}
/* SDCC per 8051 */
void timer_isr (void) interrupt 1 using 1
```

In quest'ultimo caso è stato specificato anche il set di registri alternativi da utilizzare per la ISR.
Alcuni compilatori, anche per CPU non dotate di vettorizzazione, permettono di trattare le interruzioni come se fossero vettorizzate, cioè di scrivere ISR separate per diverse sorgenti. Questo è possibile perché l'ISR vera e propria viene creata dal compilatore, e svolge la funzione di "smistare" le interruzioni:

```
/* CCS per PIC */
#INT_AD
void adc_handler(void){
…
}
#INT_RTCC NOCLEAR
void rtc_isr(void){
…
}
```

Le due ISR mostrate sono quelle dell'ADC e del timer di un PIC. Notare che nel secondo caso è stata utilizzata la keyword aggiuntiva noclear, che specifica di non resettare il flag di interruzione della periferica (quindi non specificare che l'interrupt è stato servito). In genere i compilatori dispongono di diverse keyword come queste, per accedere ai meccanismi di più basso libello.
Non è possibile entrare nei dettagli delle varianti relative a diversi compilatori ed architetture, e si rimanda per questo al manuale dello specifico compilatore utilizzato. Una cosa che possiamo notare subito è che in tutti i casi la funzione ISR non prende in ingresso ne restituisce alcun valore. Inoltre va precisato che molti compilatori traducono automaticamente la funzione ISR aggiungendo il codice necessario per salvare il contesto. I registri salvati ed il modo in cui sono salvati dipendono dal compilatore.

Uso delle interruzioni

Ora che abbiamo capito (approssimativamente) come si gestiscono le interruzioni, possiamo concentrarci sul loro utilizzo nei nostri programmi.
Per chiarire meglio le possibilità e l'utilizzo delle interruzioni consideriamo inizialmente due casi estremi: un programma sequenziale, che non usa le interruzioni, ed un programma basato solo su interruzioni. Prendiamo in considerazione un esempio abbastanza semplice: un orologio a LED, dotato di un tasto per commutare tra le 4 diverse funzioni, ed un'interfaccia seriale per potere leggere o impostare i valori da un terminale remoto.
Un tipico programma sequenziale sarà formato da un loop infinito dentro il quale saranno richiamate ciclicamente le varie funzioni di gestione (figura 1). Ad ogni ciclo verrà fatto un polling sul timer, sullo stato del tasto, sulla UART, e verrà aggiornato il display di conseguenza.

```
/* -- nel main -- */
…
while(1) {
   t=Leggi_timer();
   ...
   p=p+Leggi_stato();
   p&=3;
   …
   c=Leggi_UART();
   …
   Aggiorna_display(p);
}
```

Se le routine sono ben progettate il tutto potrebbe funzionare senza problemi, ma bastano piccoli ritardi imprevisti o calcolati male per, ad esempio, perdere un carattere ricevuto dalla UART, oppure non accorgersi di una breve pressione del tasto.
Utilizzando la filosofia diametralmente opposta si potrebbe pensare di utilizzare un loop principale "vuoto", e di gestire il sistema soltanto tramite le funzioni di interruzione. In pratica il sistema resta senza far nulla fino a quando non arriva un interrupt (figura 3). Ogni 0.5 secondi ci sarà un interrupt del timer, che servirà per aggiornare il puntino lampeggiante e dopo due passaggi l'orario visualizzato; alla pressione del tasto si aggiornerà lo stato della visualizzazione, alla ricezione di un carattere dalla UART verrà eseguita la funzione di comunicazione. Tutto questo in maniera assolutamente automatica:

```
#INT_RTCC
void Timer_int(void){
…
}
#INT_BUTTON
void Tasto_int(void){
…
}
#INT_RDA
void UART_int(void){
…
}
```

```
main() {
  …
  while(1){};
}
```

Come già detto è possibile utilizzare un simile approccio anche su processori che non hanno interrupt vettorizzati, basta che ISR principale richiami la funzione appropriate in qualche modo. Più realisticamente un programma avrà quasi sempre un loop principale in cui vengono eseguiti i compiti a bassa priorità, e delle routine di interruzione per gestire i compiti temporalmente più critici (figura 2). Rimangono comunque da analizzare alcuni aspetti poco intuitivi riguardo l'uso delle interruzioni.

All'interno di una ISR...

Come abbiamo già detto quando viene richiamata un ISR viene salvato automaticamente il Program Counter, ed i registri principali (accumulatore, status register, etc…). In genere viene data la possibilità di scegliere un set di registri alternativi, se esiste, o il compilatore stesso fa in modo da non interferire con quelli precedentemente utilizzati.
La prima cosa da fare dentro la ISR, se non la fa automaticamente il compilatore, è disabilitare le interruzioni. Questo è necessario per evitare problemi di diverso genere. Se il processore utilizzato supporta gli interrupt vettorizzati, allora è stata già richiamata la funzione appropriata, altrimenti è necessario eseguire una lettura di uno o più registri per capire quale periferica ha generato l'interruzione. A questo punto è necessario dare una conferma dell'avvenuta interruzione resettando il o i bit relativi alla periferica considerata (situati su un registro della periferica stessa o del controller delle interruzioni).

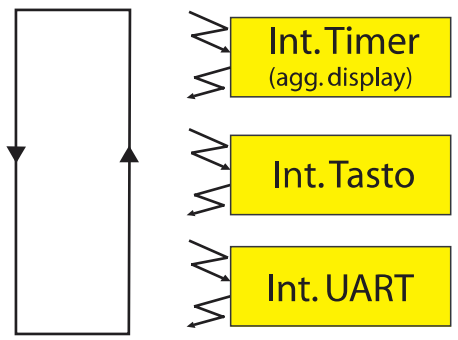


Figura 3 Programma basato solo su interruzioni per la gestione dell'orologio a LED

Dopo avere svolto i compiti richiesti, sarà sufficiente riabilitare le interruzioni (se necessario), ed uscire dalla funzione. Un esempio di ISR relativo ad un microcontrollore PIC può essere la seguente:

```
void interrupt(void) {
  if (TestBit(INTCON, T0IF)) {
    counter++;
    TMR0 = 127;
    ClearBit(INTCON, T0IF);
  }
  else if (TestBit(INTCON, RBIF)) {
    counter=0;
    TMR0 = 127;
    ClearBit(INTCON, RBIF);
  }
}
```

In questo caso si è supposto che siano abilitati soltanto due sorgenti d'interruzione: il timer TMR0 e il cambiamento sui bit della porta B, e che il compilatore si occupi del salvataggio dei registri e della disabilitazione delle interruzioni. Dal momento che i PIC non hanno un meccanismo di vettorizzazione, è necessario fare un polling: dapprima viene testato il timer, successivamente la porta B. Il registro INTCON contiene lo stato delle interruzioni, e quindi rivela quale delle due si sia verificata. Nel caso di overflow del timer, viene incrementata la variabile *counter*, caricato nuovamente il timer, e viene cancellata l'interruzione pendente. Nell'altro caso la variabile viene azzerata. I bit T0IF e RBIF del registro INTCON sono quelli corrispondenti alle sorgenti considerate, mentre le macro TestBit e ClearBit testato e resettano un determinato bit del registro specificato. Notare che per come sono scritte le due condizioni, esse sono mutuamente esclusive.
Non sempre i compiti richiesti per soddisfare un'interruzione sono eseguite dentro la ISR, questo perché potrebbe risultare troppo lento, e altre interruzioni potrebbero essere perse. A volte quindi si preferisce utilizzare funzioni esterne richiamate dal loop principale.

Passaggio di dati

Come si è notato prima le ISR non accettano ne restituiscono dati. Questo perché esse vengono richiamate improvvisamente da un punto casuale del programma, e quindi non può esserci nessu-

na routine che passa dati o prende i risultati. Per comunicare con il resto del programma si possono quindi usare dei flag o delle variabili globali, settate dalla ISR e lette dalle routine del loop principale (o viceversa).
Ritornando all'esempio dell'orologio a LED visto prima, la parte dell'ISR che gestisce la pressione del tasto potrebbe aggiornare solo la variabile relativa allo stato di visualizzazione:

```
stato=(stato++)&3;
```

In questo modo alla pressione del tasto la variabile stato assume i valori 0, 1, 2, 3, 0, 1, etc… La funzione di visualizzazione presente nel loop principale terrà conto del valore della variabile per scegliere i dati da visualizzare.
Lo stesso ragionamento può essere fatto per la gestione della UART: nella ISR relativa i dati ricevuti verranno semplicemente scritti su un buffer, sarà poi la routine apposita del loop principale che decodificherà i comandi ed invierà le risposte. In questo modo ci si assicura di non perdere i caratteri ricevuti (dal momenti che li si legge subito), e si gestisce con calma un compito poco prioritario come la decodifica delle stringhe.
Dei semplici flag possono essere invece utilizzati per cominciare al loop principale il fatto che sia avvenuta una certa interruzione, o che deve essere eseguita una funzione. Nel loop principale, tra i vari compiti da eseguire ci sarà un'istruzione tipo:

```
if (FLAGB == 1) {
  mia_funzione();
  FLAGB=0;
}
```

Sezioni critiche

Supponiamo che nel loop principale di un programma sia contenuta la seguente riga di codice (si tratta di un calcolo aritmetico relativamente intricato):

```
val=val*cost+val;
```

che succede se mentre viene eseguita questa riga di codice arriva un'interruzione, e viene eseguita la ISR? Se la ISR legge o scrive la variabile val, otterremo quasi sicuramente due risultati sbagliati! Questo succede perché l'esecuzione della riga riportata sopra non è istantanea, ma è eseguita in alcuni passi (cioè è scomposta in diverse istruzioni macchina). Se avviene un interruzione proprio nel mezzo di questi passi, da un lato la ISR leggerà un valore errato della variabile (un valore intermedio assunto durante il calcolo), dall'altro lato al ritorno la variabile sarà stata modificata, e quindi il calcolo sarà concluso utilizzando in parte un valore diverso dall'originale! In questi casi si dice che il codice in questione è una "**sezione critica**".
Esistono moltissimi casi in cui si possono avere delle sezioni critiche, oltre che nella manipolazione di variabili, anche in determinate routine complesse o legate alla comunicazione. Per evitare problemi con le sezioni critiche occorre renderle "**atomiche**", cioè bisogna assicurarsi che, una volta iniziate, esse vengano terminate prima di cedere il controllo alla ISR. Per fare questo è sufficiente disabilitare le interruzioni prima di iniziare, e riabilitarle alla fine. Alcuni compilatori hanno anche delle keyword apposite per specificare le sezioni critiche, o comunque per disabilitare ed abilitare le interruzioni.
Va notato che perfino operazioni banali come un'assegnazione tra grandezze di tipo long, eseguita su un microprocessore ad 8 bit rappresenta una sezione critica, in quanto per completare l'assegnazione occorrono 4 istruzioni di movimento ad 8 bit, e nel mezzo di queste possono verificarsi delle interruzioni. Per questo motivo, più che utilizzare la soluzione proposta prima, si preferisce di solito evitare di modificare nell'ISR le variabili modificate dal loop principale, limitandosi a leggerle, e viceversa. Tuttavia per alcuni algoritmi o procedure piuttosto articolate (in cui i dati non devono cambiare nel corso dell'esecuzione), rimane necessario rendere atomiche le sezioni interessate.
In fase di progettazione e scrittura del codice occorre prestare molta attenzione ai potenziali errori originati da sezioni critiche, perché può essere difficilissimo riuscire ad individuarli una volta che il programma è in esecuzione, dal momento che si presentano in modo sporadico, casuale, e poco ripetibile.

Interrupt multipli

Che succede se mentre si sta eseguendo una ISR sopraggiunge un'altra interruzione? Abbiamo detto, semplificando, che per evitare problemi si disabilitano le interruzioni, e si fa in modo che la ISR sia sufficientemente veloce. Questo può andare bene in molti casi, ma vale la pena considerare più in dettaglio questa situazione.

Con alcuni processori che supportano le interruzioni vettorizzate non si hanno problemi a ricevere altre interruzioni mentre si sta eseguendo una ISR, quello che succede è che si salta ad un'altra ISR, conservando il contesto. Si possono verificare anche diverse interruzioni annidate, e tutto può funzionare correttamente. Però sorge qualche problema pratico: innanzi tutto i vari contesti devono essere memorizzati nello stack, e questo di solito ha una dimensione limitata. Quindi ad un certo punto si avrà uno stack overflow. In alcuni processori, come i PIC, lo stack è implementato via hardware, ed ha una profondità molto limitata (oltre a non essere manipolabile dall'utente), quindi questa eventualità è molto più probabile. In molte applicazioni comunque si può stimare con precisione il numero massimo di interruzioni che possono arrivare, quindi accettare più interruzioni è un processo relativamente innocuo.
In altri casi, di solito quando è presente un controllore delle interruzioni, è possibile ordinare le interruzioni per priorità: in questo modo una ISR può essere interrotta soltanto da un segnale che ha priorità più elevata. Questo meccanismo limita il numero di interruzioni annidate, e permette di ottimizzare i tempi e le prestazioni. Va notate che la priorità in qualche caso può essere anche programmata via software, per cui può essere resa dinamica in funzione delle esigenze.
In altri casi ancora si ha la segnalazione di un interruzione "cumulativa" che può essere associata a diverse periferiche in attesa di risposta. Sarà la ISR a rendersi conto di questo leggendo un opportuno registro, decidendo quale soddisfare prima. Se non tutte le interruzioni pendenti sono state soddisfatte, all'uscita della ISR sarà generata una nuova interruzione, e così fino a quando tutte le periferiche non saranno state soddisfatte. L'esempio riportato prima rientra proprio in questo caso, e implementa anche via software un meccanismo di priorità tra le interruzioni: le due condizioni infatti sono scritte in modo che solo una alla volta sarà soddisfatta, ed in particolare sempre quella scritta prima.

Quando usare le interruzioni?

Le interruzioni sono il più stretto punto di collegamento tra hardware e software, per cui scegliere se, quando ed in che caso utilizzare le interruzioni, non è una questione che riguarda solamente software, ma interessa la progettazione del sistema nella sua interezza. È possibile scrivere un software interamente *interrupt driven* (basato solo sulle interruzioni) solo se l'hardware fornisce il giusto supporto. D'altro canto pur avendo a disposizioni moltissime interruzioni si può scegliere di scrivere un software che non ne faccia uso (quindi puramente sequenziale).
In alcuni casi l'uso delle interruzioni è indispensabile per ottenere buone prestazioni. Ad esempio un sistema operativo real-time dovrebbe essere basato su un segnale di temporizzazione regolare e stabile per schedulare i vari task. Oppure qualsiasi evento critico dovrebbe avere una risposta immediata da parte del software.
Si può notare anche che in alcuni casi utilizzare le interruzioni può semplificare molto la scrittura del software, semplificando il loop principale, evitando di dovere progettare in maniera accurata lo scheduling e le temporizzazioni, e rendendo più modulare il programma. Ma anche l'hardware può risultare semplificato in qualche caso, infatti è possibile delegare più compiti di controllo al software.
In qualche caso però l'uso delle interruzioni può anche presentare degli svantaggi. Un esempio si ha nel caso di processori in cui molti registri devono essere salvati: questo può rappresentare uno zoccolo di tempo sprecato ad ogni interruzione, che diventa rilevante se gli eventi si succedono con grande frequenza. In questi caso può risultare vantaggioso perfino l'uso del polling. L'altro svantaggio notevole che l'uso degli interrupt ha è la maggiore difficoltà nel debug del software. In fase di simulazione e di test molti strumenti fondamentali, come gli hardware debugger, non funzionano correttamente in presenza di interrupt, ed anche durante il normale funzionamento del sistema, individuare degli errori generati dagli interrupt risulta molto difficile, in quanto si ha un grado di osservabilità e di controllabilità notevolmente ridotto rispetto al caso di semplice codice sequenziale.

CONCLUSIONE

Molti dei concetti visti a proposito delle interruzioni verranno ripresi nella prossima conclusiva puntata di Vitamina C, in cui si parlerà dei sistemi operativi. Vedremo come funzionano, a cosa servono e come utilizzarli per rendere più efficienti i nostri programmi in C.

Le so tutte!!!

Rispondi correttamente al quiz e potrai vincere un abbonamento omaggio a

Fare elettronica

Partecipare è semplicissimo:

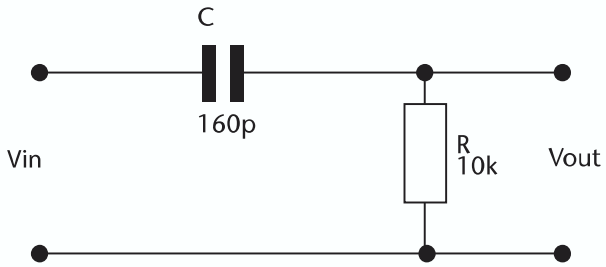
rispondi al quesito seguendo il regolamento e, se la risposta si rivelerà esatta, potrai vincere un abbonamento omaggio (o il rinnovo qualora fossi già abbonato) a Fare Elettronica. Ogni mese sulle pagine della rivista troverai la soluzione del quesito del mese precedente e il nome del vincitore di uno dei quesiti pubblicati. Per tutti i partecipanti è previsto comunque un coupon del 10%* di sconto utilizzabile per un tuo prossimo acquisto sul sito www.farelettronica.com.

* Validità 3 mesi

Quesito - LST24809

Si consideri il circuito di figura al cui ingresso Vin viene applicato un segnale ad onda triangolare simmetrico di periodo T=1ms ed ampiezza 2V picco-picco. Quali delle seguenti affermazioni sono vere?

1) L'uscita è un segnale ad onda quadra di frequenza 1KHz.
2) Il segnale di uscita ha ampiezza 2V.
3) Il segnale di uscita è periodico.
4) L'uscita è un'onda triangolare di ampiezza minore rispetto all'ingresso, ma di uguale frequenza.
5) In uscita non sarà presente alcun segnale per la presenza del condensatore in serie all'ingresso.



Scadenza: il termine ultimo per rispondere è il 30 Novembre 2005

Regolamento

1) Il quiz è aperto a tutti i lettori.
2) Saranno considerate esclusivamente le risposte pervenute entro la scadenza indicata nel quesito.
3) Inviare la risposta compilando il modulo su **www.farelettronica.com/lesotutte**.

Riflettori su... Farnell InOne:

Farnell InOne, la società internazionale di commercializzazione da Catalogo di prodotti elettronici, elettrici ed industriali e' Leader indiscusso per consegna in tutta Italia, **in sole 24 ore**, di qualsiasi prodotto elettronico. L'offerta globale di Farnell InOne porta sul tavolo di Progettisti e Buyers tutta l'elettronica di ultima tecnologia: semiconduttori, passivi, optoelettronica, cavi e connettori, elettrotecnica, strumenti di misura, automazione industriale, pneumatica, sensori, utensili e meccanica.
Grazie agli altissimi investimenti nell'ingegnerizzazione dei processi logistici e nello stock dei prodotti, Farnell InOne, supportata dalle enormi potenzialità di una multinazionale come il Gruppo Premier Farnell, con 5000 dipendenti in oltre 20 nazioni, è in grado di garantire la certezza di ricevere il giorno successivo all'ordine qualsiasi prodotto scelto fra i **180.000** a disposizione e immediatamente ordinabile dal Catalogo, dal sito web www.farnellinone.it o dal CD-rom.

IL PIÙ PERFORMANTE E-COMMERCE DELL'ELETTRONICA!

Con **Electronic Design World** abbiamo creato un "sito nel sito", realizzato specificatamente per gli Electronic Designers - commenta l'Ing. Stefano Noseda, General Manager di Farnell InOne Italia dal 2003.
Al suo interno gli ultimi aggiornamenti dal mondo dell'elettronica su prodotti, tecnologie e trend del settore ed, inoltre, un innovativo "FORUM DI DISCUSSIONE" aperto a chi intende scambiare opinioni su tutte le tematiche del Design Engineering.
Le potenzialità del nuovo sito sono amplificate con le ***funzionalità*** del E-Commerce.
Innanzitutto i numeri: oggi solo dal sito WEB l'utente italiano può ordinare oltre **180.000** prodotti con 14.000 novità e questa offerta rappresenta la più vasta del settore.
Il nuovo sito permette di verificare in tempo reale la disponibilità della merce a magazzino, di programmarsi le consegne, verificare lo status dell'ordine ed ottenere il numero della lettera di vettura per verificarne la consegna "step by step".
Dotato di un motore di ricerca di ultima generazione e potente quanto a velocità di indicizzazione, il sito è caratterizzato dalla semplificazione dei processi di ricerca prodotti: i risultati vengono automaticamente ripartiti per categorie permettendo di circoscrivere l'ambito della ricerca per poi affinarla.
Altra caratteristica innovativa è la possibilità di effettuare ricerche di prodotti similari in base alle caratteristiche parametriche degli stessi per giungere poi a quanto di più interessante si richiede a qualsiasi catalogo elettronico e cioè la COMPARAZIONE DEI PRODOTTI SELEZIONATI.
Il nuovo WEB dispone, poi, di un sistema multiutente, progettato soprattutto per i Buyers, che permette di gestire i profili di più utenti secondo livelli di creazione, modifica e controllo delle "permission". Si tratta di una soluzione che fornisce, a costo zero, supporto ai responsabili acquisti dei clienti più evoluti nella gestione interna dell'ufficio approvvigionamenti.
Questo concentrato di informazioni - continua Noseda - ha permesso di imporci, all'interno dello scenario della distribuzione da Catalogo, non solo per il servizio di consegna in 24 ore, ma per una più qualificata offerta rivolta, in modo dedicato, alla categoria dei Progettisti Elettronici, che rappresentano la maggioranza dei nostri clienti.

FARNELL INONE: LEADER NEL PROMUOVERE IL RECEPIMENTO DELLE DIRETTIVE ROHS

Il 1 luglio 2006 segna l'entrata in vigore delle Direttive Europee che limitano l'uso di determi-

Tutta l'Elettronica in sole 24 ore

nate sostanze pericolose nelle apparecchiature elettriche ed elettroniche (RoHS e WEEE).
Come noto la Comunità Europea ha pubblicato, ormai da tempo, le norme entro le quali gli operatori del settore dovranno agire in qualità di produttori e/o distributori ed, in tal senso, Farnell InOne si è messa in prima linea, svolgendo, ormai da 2 anni, un ruolo di primissimo piano nell'erogazione di informazioni e assistenza a clienti e fornitori.

Vogliamo essere identificati dagli utenti come PUNTO DI RIFERIMENTO per qualsiasi esigenza per diventare il DISTRIBUTORE DI FIDUCIA in ambito ROHS - dice Noseda. La nostra *mission*? - continua l'Ing. - abbracciare lo spirito delle iniziative verdi favorendo il miglioramento delle performance aziendali e offrendo vantaggi concreti ai clienti e all'ambiente.
Grazie a questo impegno, Farnell inOne detiene oggi una posizione di primato in materia RoHS e WEEE e tanti sono ormai i risultati concreti. Innanzitutto è stato creato all'interno dell'E-Commerce di Farnell InOne, un vero e proprio portale dedicato alle Direttive RoHS & WEEE, *www.rohs.info*, dal quale si possono scaricare i testi delle normative, le più frequenti Domande e Risposte ed i suggerimenti degli opinion leader.

Molteplici sono poi le pubblicazioni. Farnell InOne, in collaborazione con ERA Technology, società di consulenza della CE, ha sviluppato, alla fine dell'anno scorso, la **PRIMA GUIDA in EUROPA** sulla restrizione all'uso di alcune sostanze ritenute pericolose ed inquinanti nei materiali elettrici ed elettronici, immediatamente scaricabile dal sito *www.farnellinone.it*. Tale Guida è stata poi integrata in un nuovo catalogo dedicato ai Prodotti Conformi, con oltre 1500 prodotti "compatibili RoHS". Altri gli strumenti di informazione, fruibili anche dal sito: un vero e proprio **"MANUALE TECNICO LEGISLATIVO"** che comprende un approfondimento sugli aspetti inerenti la saldatura senza-Piombo e una **BROCHURE DIVULGATIVA** con inserti dedicati ai progettisti, ai produttori CEM, agli uffici acquisti e ai reparti di manutenzione.

Con il mese di Settembre 2005 arriva il primo e unico **MAXI-Catalogo RoHS**, con oltre 60.000 prodotti conformi. È stato un lavoro lunghissimo di analisi e certificazioni, ma il risultato è entusiasmante - dice Noseda - e, a testimonianza di ciò, gli oltre 25.000 utenti in Italia che, da prima dell'estate, hanno prenotato una copia di questo primo listino RoHS.
Conclude Noseda - Sono tutti investimenti necessari per qualificare la nostra offerta su un piano tecnologico diverso rispetto al tradizionale distributore da catalogo. Vogliamo trasmettere sicurezza ai nostri Clienti. E solo con un'attenta cura per le nuove tecnologie, le normative internazionali, che cambiano ed impongono agli operatori del settore una continua rivisitazione dei processi produttivi e distributivi, è possibile affermare ogni giorno il proprio posizionamento competitivo nel mercato.

Per ricevere GRATUITAMENTE il Catalogo Generale o le ultime pubblicazioni RoHS contattare il Call Center di Lainate al numero 02.93995200 o all'indirizzo: *vendite@farnellinone.com*

Riflettori su...

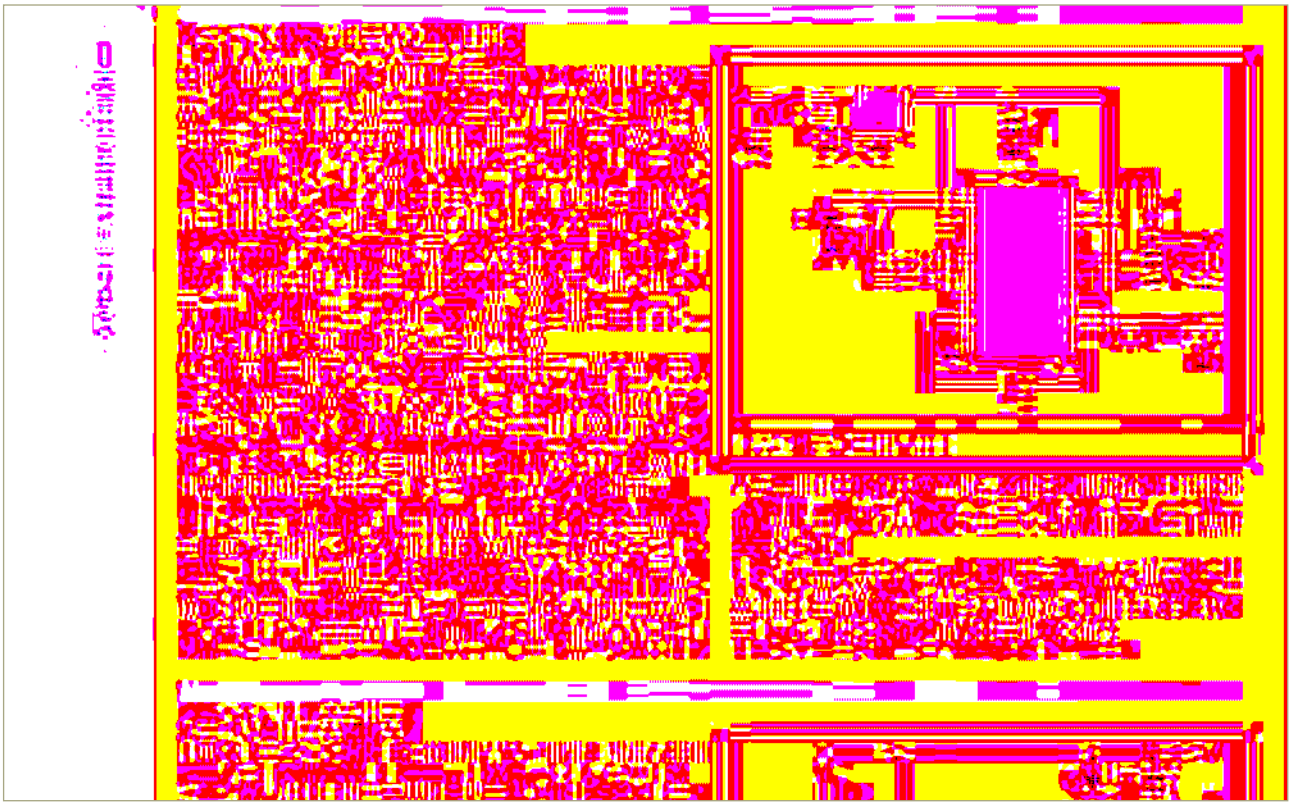
Numero Speciale:

Come ampiamente anticipato, questo mese Fare Elettronica è in edicola con un bellissimo numero speciale, **in omaggio agli abbonati**: *100 idee di progetto per 1000 applicazioni.*
Una serie di progetti utili per la risoluzione dei comuni problemi di progettazione: pilotare un interruttore con un raggio luminoso, costruire un timer, realizzare un telecomando a radiofrequenza o ad infrarossi, costruirsi una serie di strumenti da laboratorio…
Questi sono solo alcuni degli argomenti trattati in questo numero speciale dedicato ai lettori di Fare Elettronica e a tutti coloro che necessitano di idee, spunti o approfondimenti sulle applicazioni dei componenti elettronici.
I progetti raccolti non hanno la presunzione di costituire sempre una soluzione pronta all'uso, ma hanno lo scopo di illustrare idee e tecniche circuitali da usare in progetti più ampi ed ambiziosi.
Volutamente tutti i progetti non utilizzano circuiti integrati particolari, ma si basano prevalentemente su componenti discreti o circuiti integrati comuni e di facile reperibilità.
Questa caratteristica fa di questo fascicolo speciale, un utile prontuario sempre attuale che si guadagnerà sicuramente un posto d'onore nella libreria del vostro laboratorio.
Nella pagina accanto trovate un esempio dei cento circuiti contenuti nello speciale.

I progetti sono suddivisi in sei diverse categorie: *Video, Audio, Sensori e strumentazione, Radiantistica e telecomandi, Alimentatori e caricabatterie, Sistemi di controllo, Timers e vari.*
Ciascuna categoria è contrassegnata da uno specifico colore e questo accorgimento faciliterà notevolmente la consultazione permettendovi di trovare immediatamente la soluzione al vostro problema.

Lo speciale "100 idee di progetto per 1000 applicazioni" vi attende in edicola al prezzo di € 6,00. Non lasciatevelo sfuggire!

100 idee di progetto per 1000 applicazioni



Sì approfitto della fantastica occasione e mi abbono a 11 numeri di Fare Elettronica e pagherò solo € 45,00

DATI PERSONALI

Nome .....

Cognome .....

Via ..... n° ....

Cap ..... Città ..... Prov ....

Tel ..... Fax .....

Email .....

Ragione Sociale .....

P.Iva ..... ❍ Ricevuta

Privacy. Ai sensi del Decr. Lgs. 196/2003 la informiamo che i dati trasmessi verranno impiegati coi principali scopi di indagini di mercato e nelle modalità previste dallo stesso, prevalentemente con mezzi informatici. Il conferimento, di norma facoltativo, è obbligatorio per permettere il rapporto commerciale. È in ogni caso fatto diritto dell'interessato esercitare i propri diritti, nei modi previsti dal "Titolo II art. 7" della legge sopra citata, scrivendo a Inware Edizioni Via Cadorna 27 – 20032 Cormano o trmite email a info@inwaredizioni.it

ABBONATI SUBITO

Compila, ritaglia e spedisci via fax questo coupon allo **02-66508225**

Spedisci questa pagina in busta chiusa a:
INWARE Edizioni
Via Cadorna, 27/31 - 20032 Cormano (MI)

Chiamaci allo **02-66504794**

Abbonati on-line sul sito
www.farelettronica.com/abbonamento

MODALITÀ DI PAGAMENTO

❍ **CARTA DI CREDITO**

❑ American Express ❑ Visa ❑ Mastercard

Titolare .....

n° ..... scad .....

❍ **VERSAMENTO SUL CCP 22790232**

Allegare la ricevuta (o copia) del versamento intestato a Inware Srl, indicando nella causale: "Abbonamento Fare Elettronica"

❍ **BONIFICO BANCARIO**

Appoggiarlo su: **Poste Italiane - CIN: Z - ABI: 07601 CAB: 01600 - C/C: 000022790232** intestato ad **Inware srl**

❍ **ALLEGO UN ASSEGNO**

intestato a Inware Srl

Firma .....

Abbonati subito!

- ✔ **Risparmierai ben 15,50 euro sul prezzo di copertina**
- ✔ **Avrai la garanzia del prezzo bloccato per un anno**
- ✔ **La rivista ti sarà recapitata comodamente a casa**
- ✔ **Compreso con l'abbonamento (o il rinnovo) riceverai un buono sconto del 20%* per un tuo prossimo acquisto sul sito www.farelettronica.com**
- ✔ **Compila il cedolino qui a fianco e invialo in busta chiusa o via fax, riceverai un gradito regalo a sorpresa**

* Validità 3 mesi

INWARE EDIZIONI